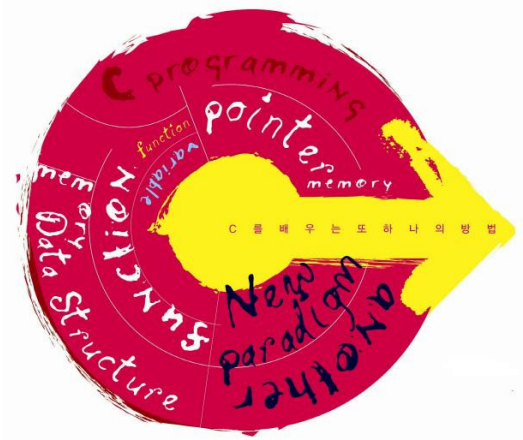
Clickseo.com

C Programming

제어 흐름

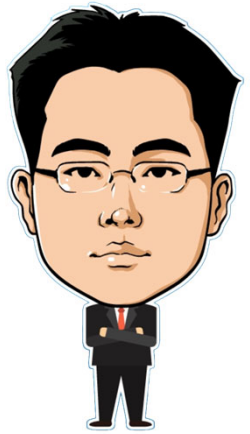
(Control Flow)



Seo, Doo-Ok

Clickseo.com

clickseo@gmail.com



CC BY NC ND

목 차

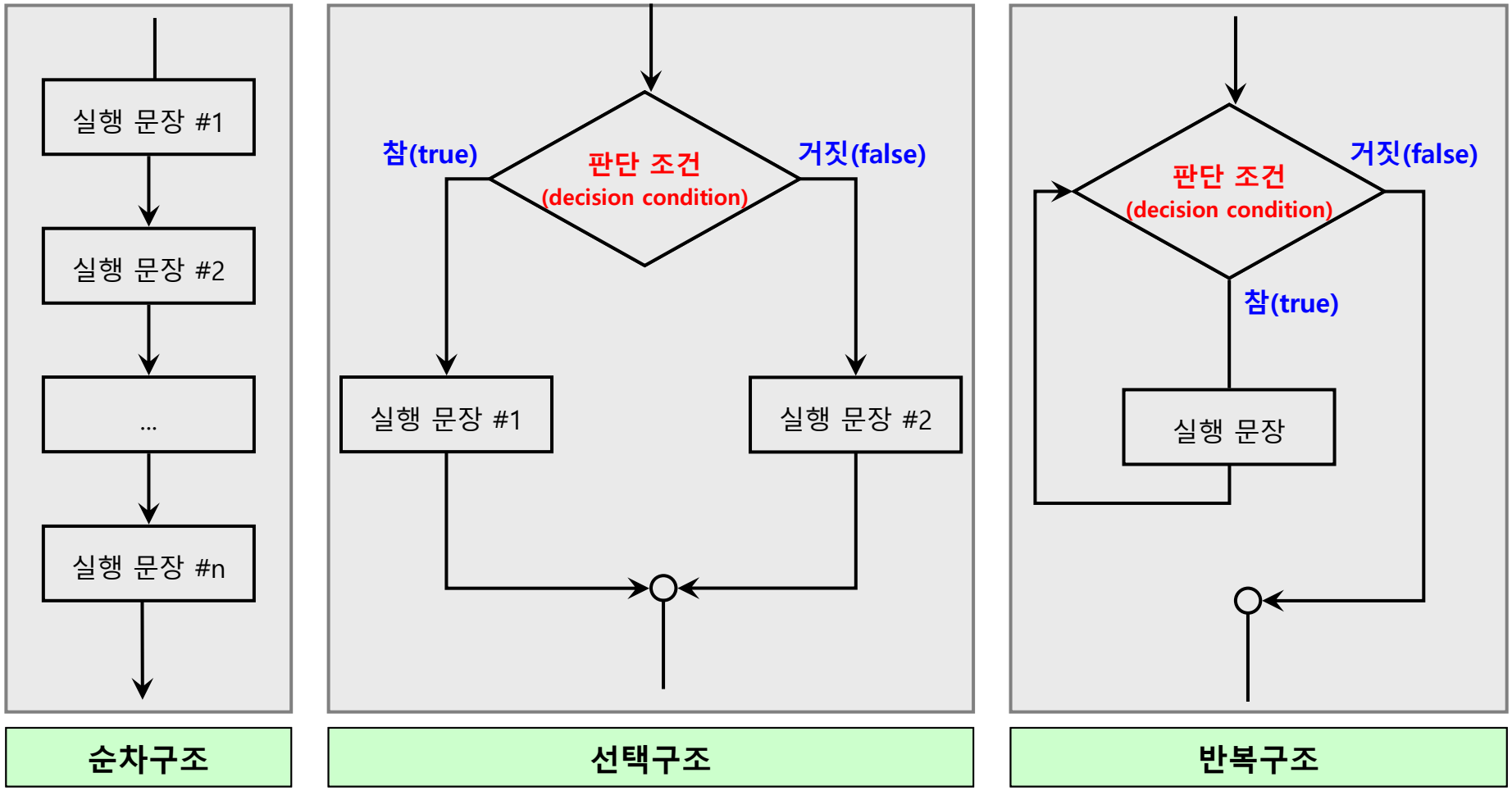
CC BY NC ND

백문이불여일타(百聞而不如一打)

- 선택 구조
- 반복 구조
- 점프문

제어흐름 (1/2)

- **순서도(Flowchart): 알고리즘을 그림으로 표현**



제어흐름 (2/2)

- **의사코드(pseudo-code): 영어와 비슷한 자연어로 표현**
  - **특정 프로그래밍 언어의 문법에 따라 쓰인 것이 아니라, 일반적인 프로그래밍 언어와 형태가 유사하다.**
    - 특정 언어로 프로그램을 작성하기 전에 알고리즘의 모델을 대략적으로 모델링

순차구조	선택구조	반복구조

action 1
action 2
.
.
.
action n
	if (condition)
	then
	action
	else
	action
	End if
		while (condition)
		action
		action
		End while

선택 구조

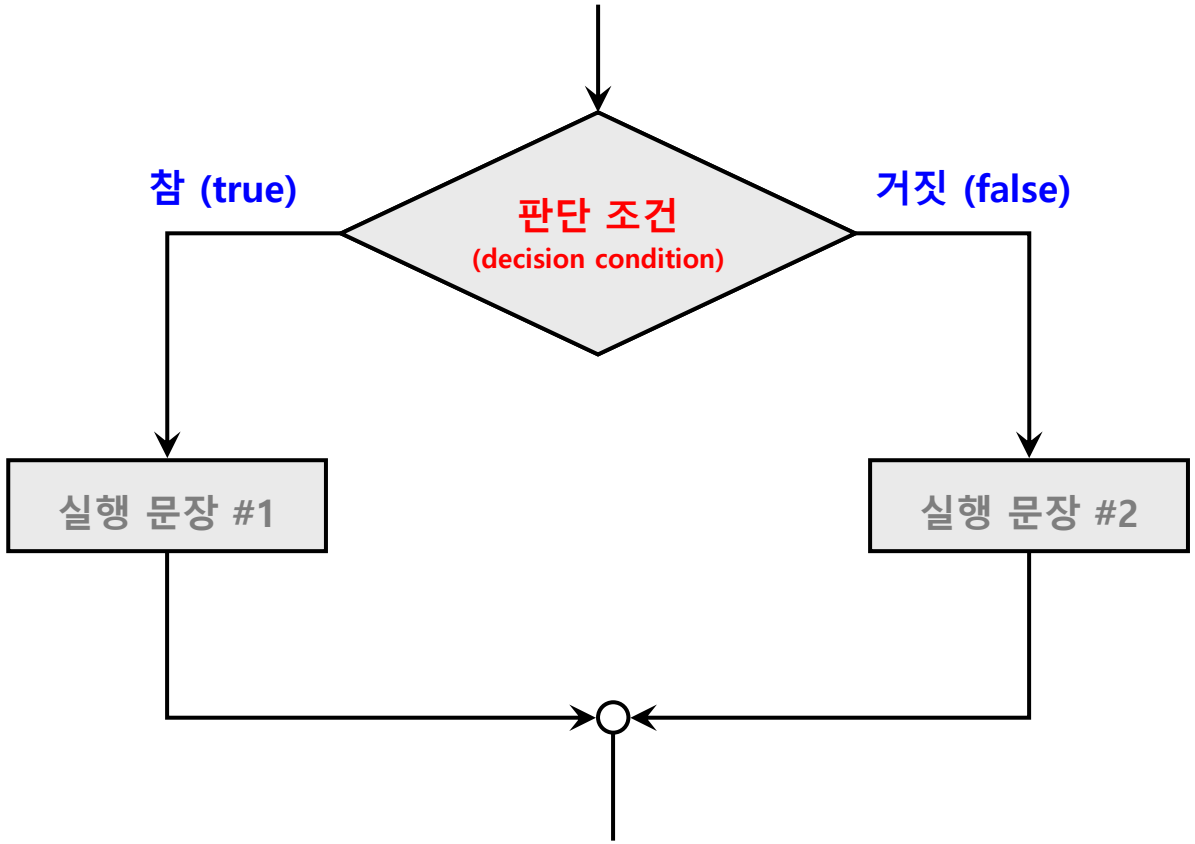
CC BY NC ND

- **선택 구조**
  - 이중 선택
  - 다중 선택
- 반복 구조
- 점프문

백문이불여일타(百聞而不如一打)

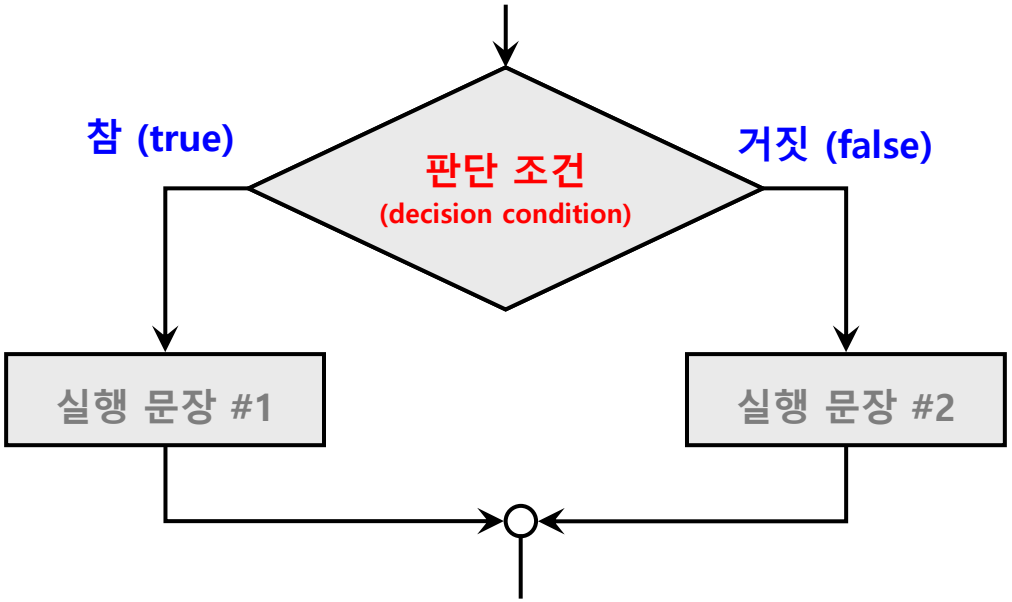
이중선택 (1/6)

- **이중선택**(Two-Way Selection)
  - 참이나 거짓 중 하나의 문장 만을 실행



이중선택 (2/6)

- **if-else**
  - 두 가지 선택 사이에서 결정을 내리기 위해 사용되는 구조
    - 두 문장이 동시에 실행되는 것은 불가능




```
if ( 판단조건 )
        실행 문장 #1
else
        실행 문장 #2
```

이중선택 (3/6)

```
if ( i == 5 )          temp++;
else                   temp--;
```

단순한 if-else 문

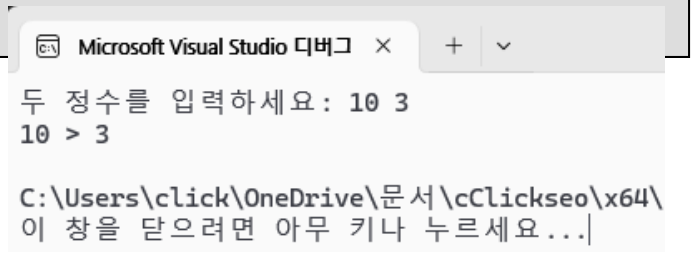
```
if ( i == 5 ) {
            temp++;
            printf("%d",  temp);
}
else {
            temp--;
            printf("%d",  temp);
}
```

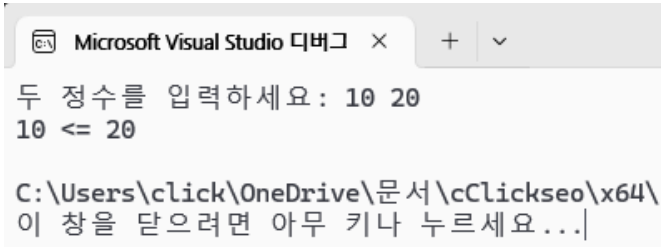
if-else 안에 있는 복합문

이중선택 (4/6)

예제 3-1: 이중 선택 -- if-else

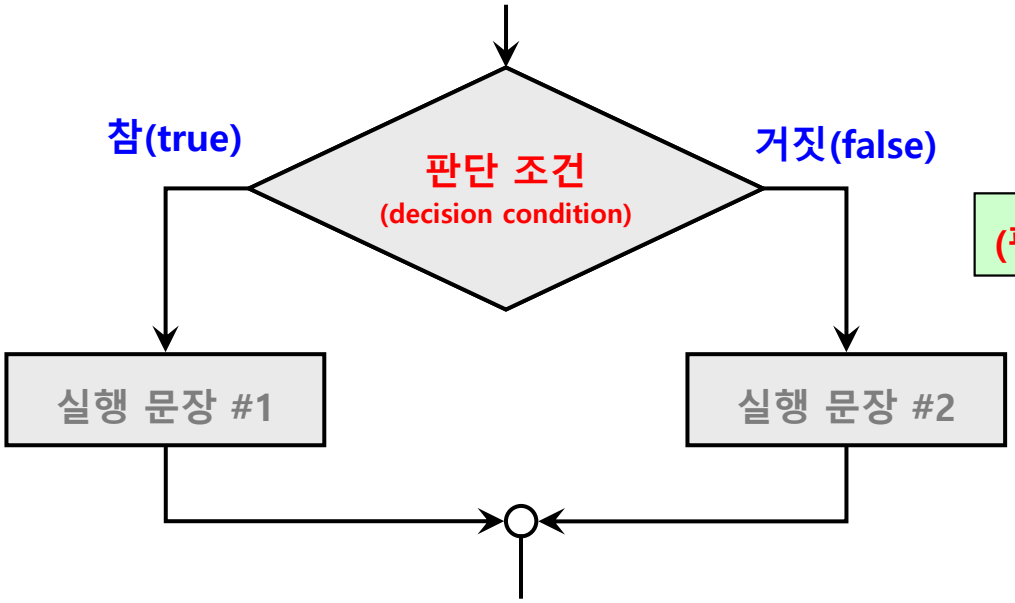
```
#include <stdio.h>
int main(void)
{
        int     a, b;

        printf("두 정수를 입력하세요: ");
        scanf_s("%d %d", &a, &b);     // scanf("%d %d", &a, &b);

    if ( a <= b )       printf("%d <= %d₩n", a, b );
    else                printf("%d > %d₩n", a, b );

        return 0;
}
```





이중선택 (5/6)

- **삼원 조건식: 조건 연산자**
  - 전통적인 if-else 에 대하여 편리한 대안을 제공
  - 세 개의 피연산자와 두 개의 연산자를 갖는다.
    - 먼저 가장 왼쪽의 수식을 평가
      - 왼쪽의 수식이 **참:** 실행 문장 #1
      - 왼쪽의 수식이 **거짓:** 실행 문장 #2



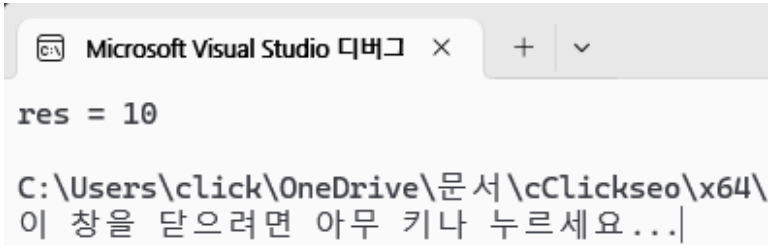
(판단 조건) ? 실행 문장 #1 : 실행 문장 #2

이중선택 (6/6)

예제 3-2: 이중 선택 -- if-else 와 삼원 조건식

```c
#include <stdio.h>

int main(void)

{

        int a = 10, b = 20, res;

        if( a < b )         res = a;
        else                res = b;

        printf("res = %d \n", res);


        return 0;

}
```

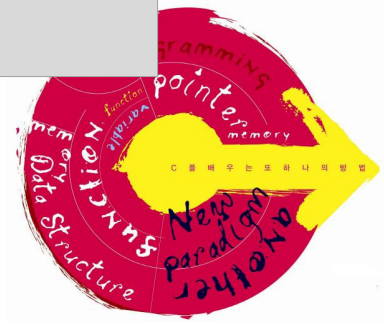
```
res = ( a < b ) ? a : b;
```



CC BY NC ND

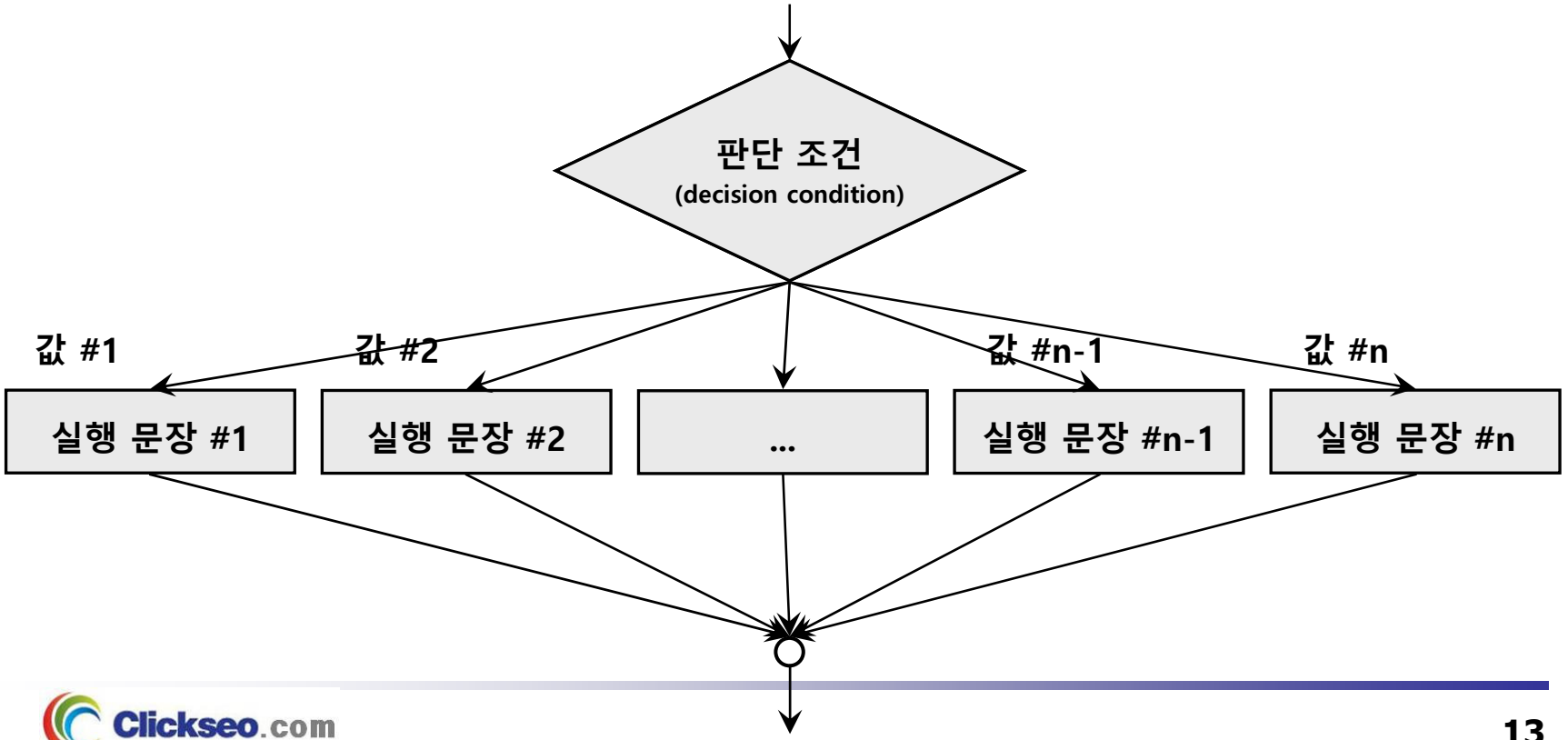
선택 구조

다중 선택



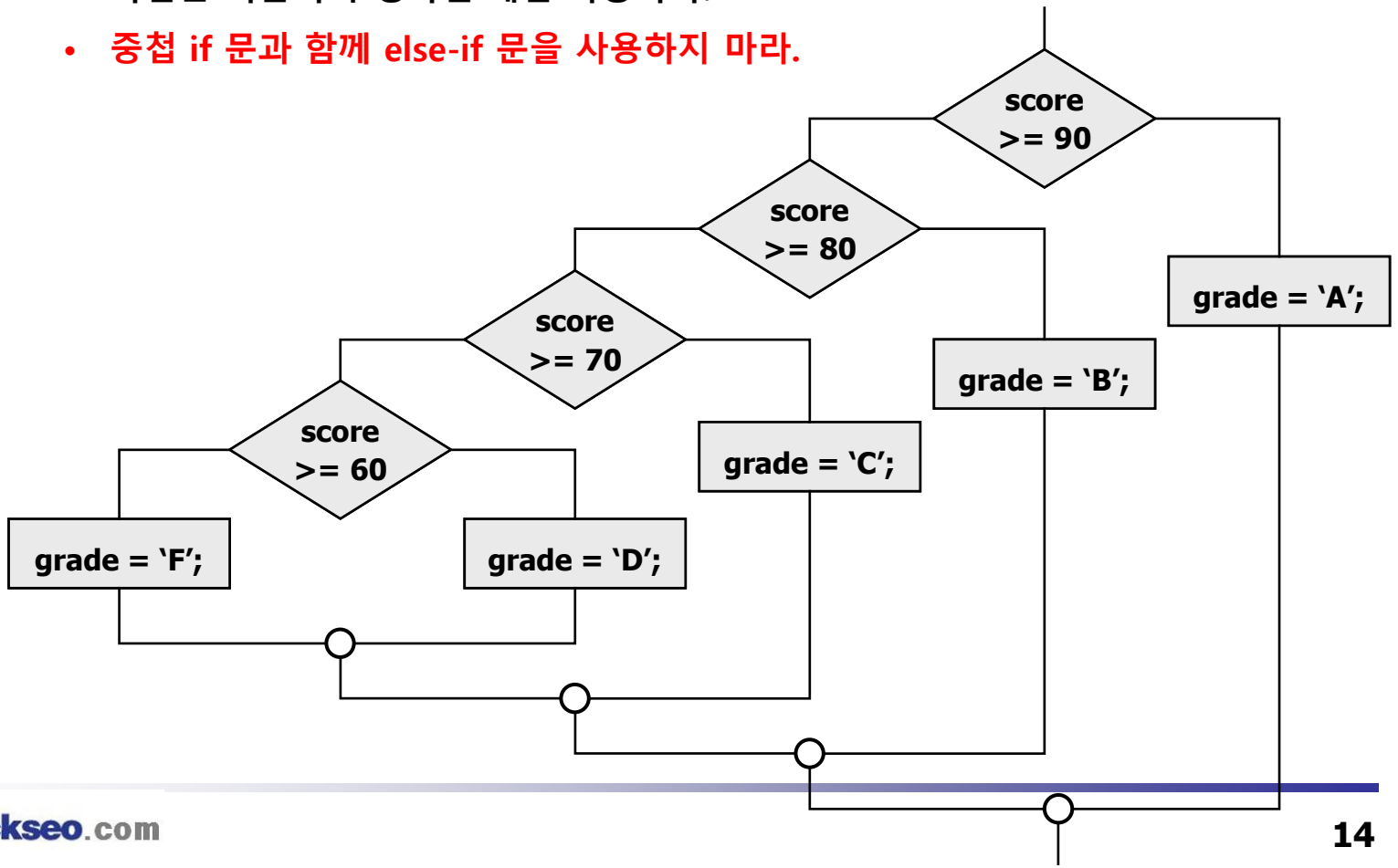
다중선택 (1/5)

- **다중선택**(Multiway Selection)
  - 여러 대안 중에서 선택을 하는 것이다.
    - **else-if** 문: 주어진 범위의 값에 기초할 때 사용(정수가 아닌 값)
    - **switch** 문: 선택 조건이 정수식으로 정리될 때 사용



다중선택 (2/5)

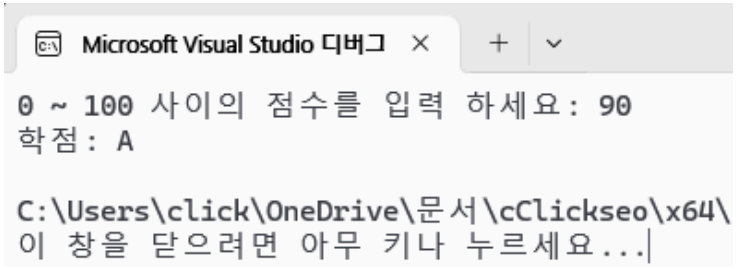
- **else-if 문**
  - 정수가 아닌 값에 기초하여 다중결정을 해야 할 때 사용
    - 똑같은 기본식이 평가될 때만 사용하라.
    - 중첩 if 문과 함께 else-if 문을 사용하지 마라.



다중선택 (3/5)

예제 3-3: 다중 선택 -- else-if 문

```c
#include <stdio.h>
int main(void)
{
        int  score;
        char grade;

        printf("0 ~ 100 사이의 점수를 입력 하세요: ");
        scanf_s("%d", &score);        // scanf("%d", &score);

        if(score >= 90)          grade = 'A';
        else if(score >= 80)     grade = 'B';
        else if(score >= 70)     grade = 'C';
        else if(score >= 60)     grade = 'D';
        else                     grade = 'F';

        printf("학점: %c \n", grade);

        return 0;
}
```



다중선택 (4/5)

- **switch 문**
  - 많은 대안들 중에서 하나의 결정을 내리기 위해서 사용하는 복합문

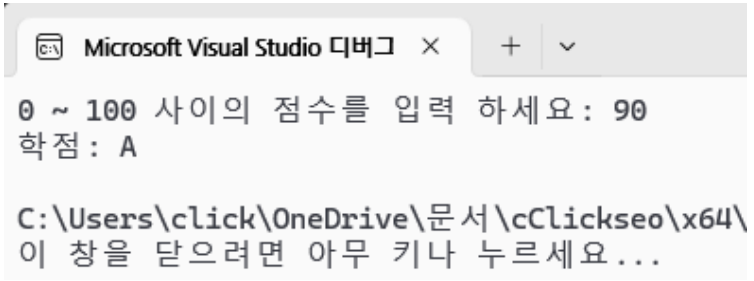
```
switch ( 표현식 )
{
        case 정수형 상수:  실행 문장;
                           실행 문장;
        case 정수형 상수:  실행 문장;
                           실행 문장;
        case 정수형 상수:  실행 문장;
                           실행 문장;
        default: 실행 문장;
}
```

"선택 조건은
정수 형태들 중의 하나이다."

다중선택 (5/5)

예제 3-4: 다중 선택 -- switch 문

```c
#include <stdio.h>
int main(void)
{
        char    grade;
        int     score, temp;

        printf("0 ~ 100 사이의 점수를 입력 하세요: ");
        scanf_s("%d", &score);        // scanf("%d", &score);

        temp = score / 10;
        switch(temp) {
                case 10:
                case 9: grade = 'A';        break;
                case 8: grade = 'B';        break;
                case 7: grade = 'C';        break;
                case 6: grade = 'D';        break;
                default: grade = 'F';
        }
        printf("학점: %c \n", grade);

        return 0;
}
```

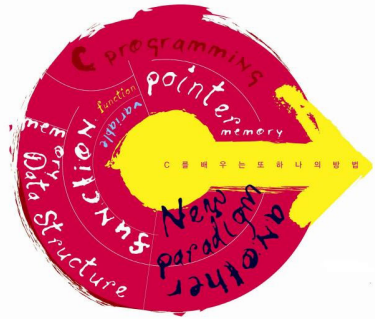


반복 구조

CC BY NC ND

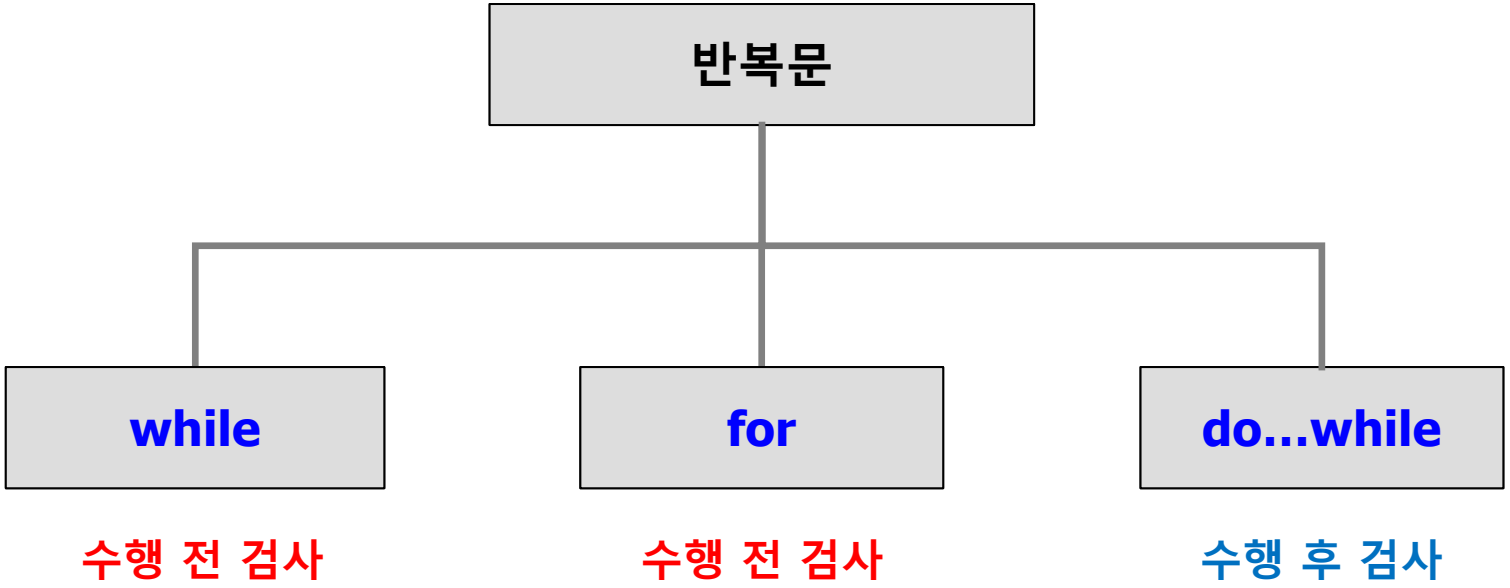
백문이불여일타(百聞而不如一打)

- 선택 구조
- **반복 구조**
  - 반복문
- 점프문



반복 구조

- **반복문**



반복문 (1/8)

- **while 문**
  - 사용 예

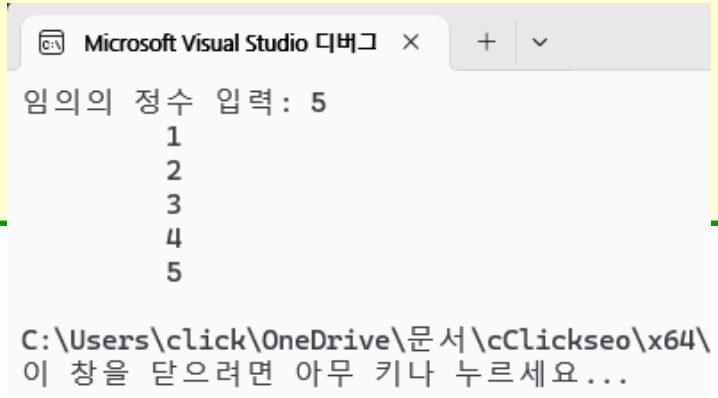
```
while ( 조건식 )
{
    …
    ( 실행할 명령문 );
    …
}
```

```
#include <stdio.h>
int main(void)
{
        int         i = 0;      // 초기문: 조건 초기화
        while ( i < 10 )        // 조건문: 판단 조건
        {
                printf("Hello World!!! ₩n");
                i++;            // 증감문: 조건의 변화
        }

        return 0;
}
```

반복문 (2/8)

예제 3-5: while 문을 이용한 수열 출력

```
#include <stdio.h>
int main(void)
{
        int     limit;

        printf("임의의 정수 입력: ");
        scanf_s("%d", &limit);          // scanf("%d", &limit);

        int     i = 1;
        while( i <= limit ) {
                printf("₩t %d ₩n", i);
                i++;
        }

        return 0;
}
```



반복문 (3/8)

for 문

- 괄호 안에 3개의 수식을 포함한다.
  - 첫 번째 수식: **초기문** (조건의 초기화)
  - 두 번째 수식: **조건문** (판단 조건)
  - 세 번째 수식: **증감문** (조건의 변화)

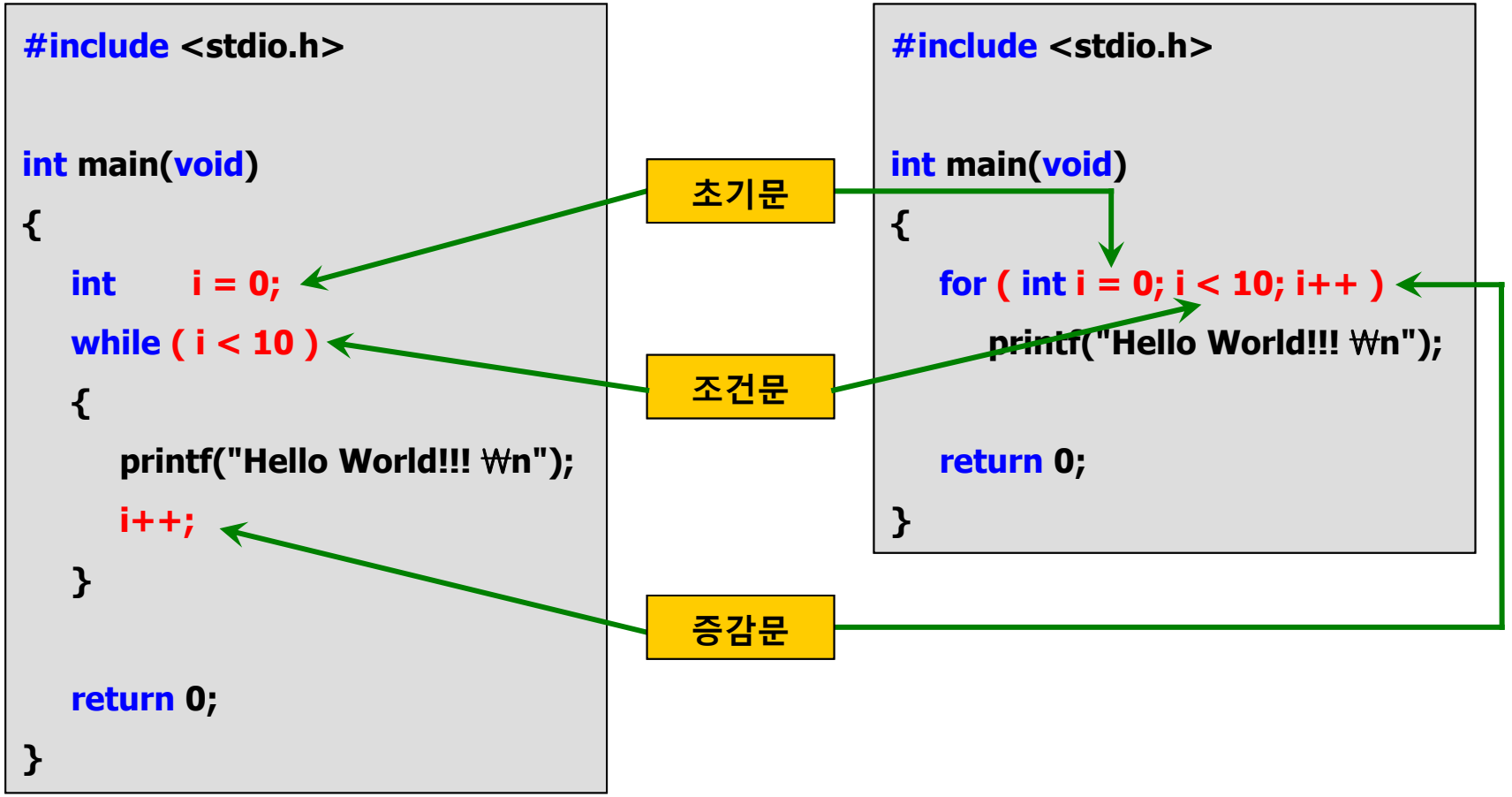
```
for ( 초기문; 조건문; 증감문 )
{
        …
        실행할 명령문;
        …
}
```

- while문과 같이 세미콜론(;)이 붙지 않는다.

- **중첩 for 문:** 하나의 for문 또한 for문의 몸체에 위치할 수 있다.

반복문 (4/8)

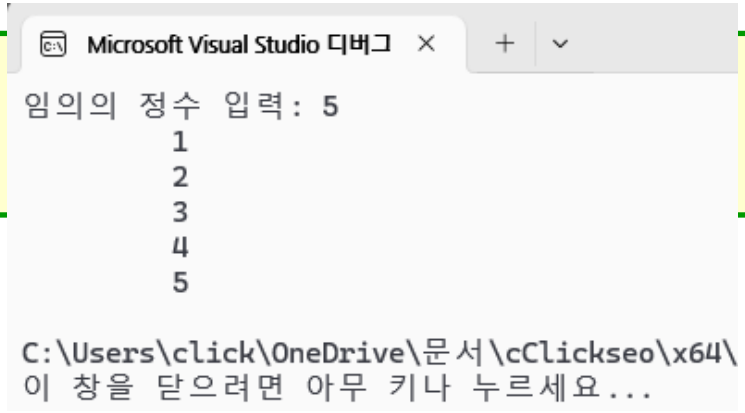
- for 문과 while 문



반복문 (5/8)

예제 3-6: for 문을 이용한 수열 출력

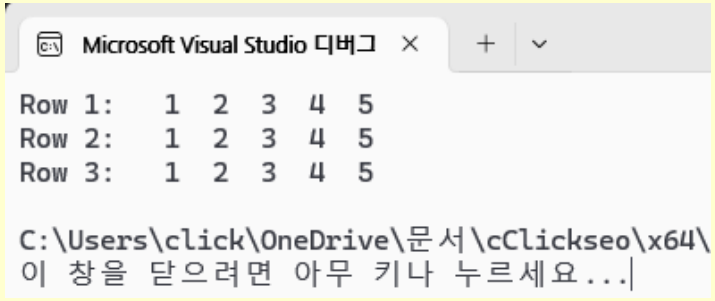
```
#include <stdio.h>

int main(void)

{

        int     limit;


        printf("임의의 정수 입력: ");

        scanf_s("%d", &limit);        // scanf("%d", &limit);

        for ( int i = 1; i <= limit; i++ )

                printf("₩t %d ₩n", i);


        return 0;

}
```



반복문 (6/8)

예제 3-7: 중첩 for 문

```
#include <stdio.h>

int main(void)

{
    for ( int i = 1; i <= 3; i++ ) {
            printf("Row %d: ", i);
            for ( int j = 1; j<= 5; j++ )
                    printf("%3d", j);
            printf("₩n");
    }

    return 0;

}
```



반복문 (7/8)

- **do-while 문**
  - 수행 후 검사
    - 세미콜론(;)으로 끝난다.

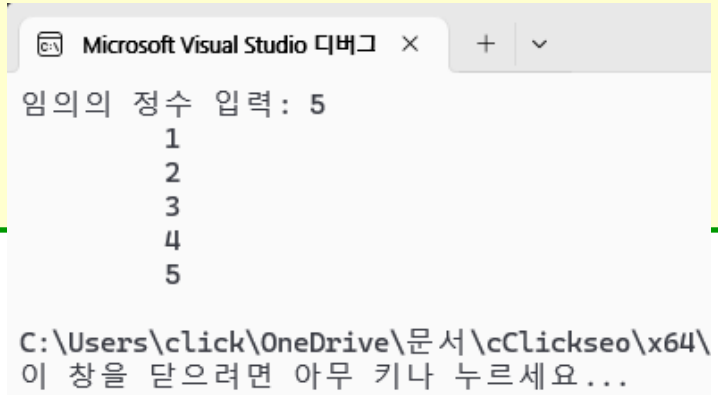
```
do
    ( 실행할 명령문 )
while ( 조건식 );
```

```
do {
    ...
    실행할 명령문;
    ...
}
while ( 조건식 );
```

반복문 (8/8)

예제 3-8: do-while 문을 사용하여 누적 합 계산

```
#include <stdio.h>
int main(void)
{
        int     limit;

        printf("임의의 정수 입력: ");
        scanf_s("%d", &limit);          // scanf("%d", &limit);

        int     i = 1;
        do {
                printf("₩t %d ₩n", i);
                i++;
        }while( i <= limit );

        return 0;
}
```



점프문

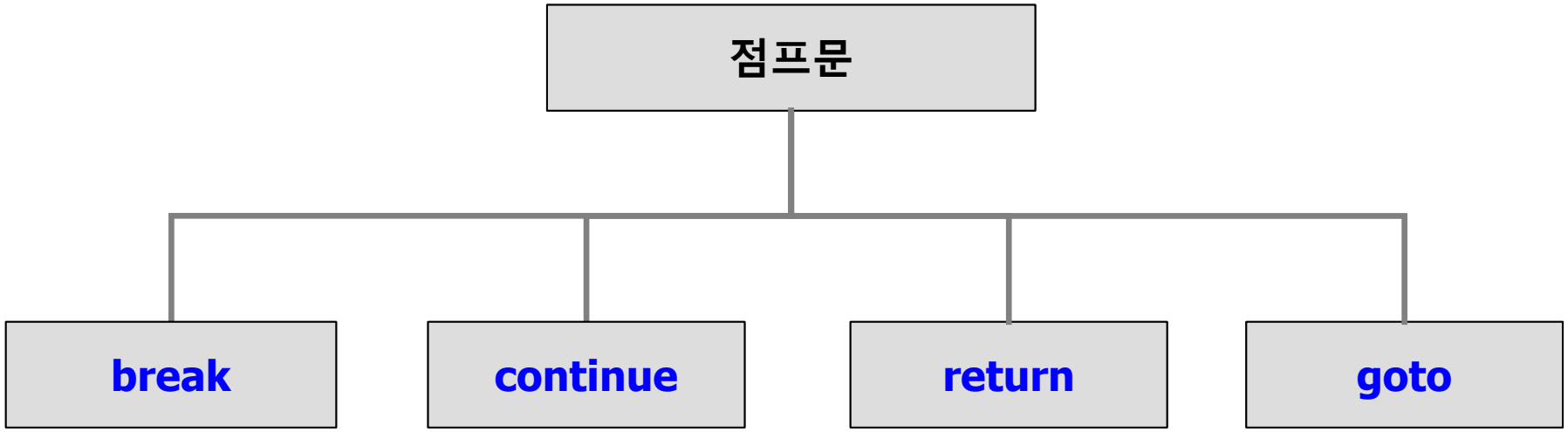
CC BY NC ND

백문이불여일타(百聞而不如一打)

- 선택 구조
- 반복 구조
- **점프문**

점프문 (1/6)

- **점프문**(Jump Statement)
  - goto 문은 구조적 프로그래밍에 적합하지 않다.



점프문 (2/6)

● break

❍ 반복문을 종료

- 반복문의 조건 검사를 거짓(false)으로 하는 것과 동일한 의미이다.
- "만약 중첩된 반복문을 사용한다면, 현재 위치한 가장 내부의 반복문 만을 종료."

```
#include <stdio.h>
int main(void)
{
    for(int i=1; i<=5; i++)    {
            if( i==3 )
                    break;

            printf("i: %d \n", i);
    }

    printf("프로그램 종료!!! \n");

    return 0;
}
```

```
Microsoft Visual Studio 디버그
i: 1
i: 2
프로그램 종료!!!

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

점프문 (3/6)

- **continue**
  - 반복문을 종료 시키는 것이 아니라, 검사식(while, do-while)이나 변경식(for)으로 제어가 이동한다.

```
#include <stdio.h>
int main(void)
{
    for(int i=1; i<=5; i++)    {
        if(i==3)
            continue;

        printf("i: %d \n", i);
    }

    printf("프로그램 종료!!! \n");

    return 0;
}
```

```
Microsoft Visual Studio 디버그
i: 1
i: 2
i: 4
i: 5
프로그램 종료!!!

C:\Users\click\OneDrive\문서\cClickseo\x64\
이 창을 닫으려면 아무 키나 누르세요...
```

점프문 (4/6)

- **goto**
  - 중첩된 반복문에서 한번에 모든 반복문을 벗어날 경우 사용한다.
    - break문은 한번에 한 개의 루프만 벗어날 수 있다.
    - 에러가 여러 곳에 걸쳐서 발생할 수 있을 때 유용하다.

- **Label**
  - lable문 뒤에는 콜론(:)이 붙는다.

```
for ( ... ) {
    for ( ... ) {
        ...
        if ( ... )
            goto error;
    }
    ...
}

error:
        실행 문장;
```

점프문 (5/6)

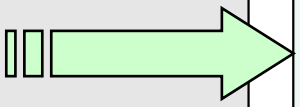
● 구조적 프로그래밍 기법

❍ 좋은 구조적 프로그래밍은 break 문의 사용을 제한한다.

- 다음과 같은 예제는 어떠한 반복문내에서도 사용되지 않는 것이 좋다.

비구조적 프로그래밍

```
for ( ; ; ) {
    ...
    if (state)
        break;
}
```

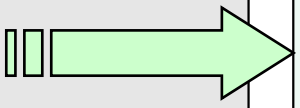


권장!!!

```
for ( ; !state ; ) {
    ...
}
```

비구조적 프로그래밍

```
while ( x ) {
    ...
    if (state)
        break;
    else
        ...
}
```



권장!!!

```
while ( x && !state ) {
    ...
    if ( !state )
    ...
}
```

점프문 (6/6)

- **구조적 프로그래밍 기법:** 상태 플래그
  - 상태 플래그 사용
    - 제한 조건이 너무 복잡한 경우 상태 플래그가 사용될 수 있다.

```
state = 0;
while ( !state ) {
        ...
        if ( x == 5 )
                state = 1;
        else
        ...
}
```

참고문헌

[1] 서현우, “혼자 공부하는 C 언어: 1:1 과외 하듯 배우는 프로그래밍 자습서”, 한빛미디어, 2023.

[2] Paul Deitel, Harvey Deitel, "C How to Program", Global Edition, 8/E, Pearson, 2016.

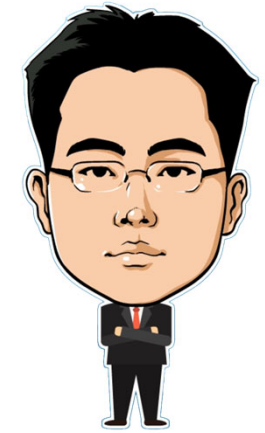
[3] Kamran Amini, 박지윤 번역, “전문가를 위한 C : 동시성, OOP부터 최신 C, 고급 기능까지!”, 한빛미디어, 2022.

[4] 서두옥, “(열혈강의) 또 하나의 C : 프로그래밍은 셀프입니다”, 프리렉, 2012.

[5] Behrouz A. Forouzan, Richard F. Gilberg, 김진 외 7인 공역, “구조적 프로그래밍 기법을 위한 C”, 도서출판 인터비젼, 2004.

[6] Brian W. Kernighan, Dennis M. Ritchie, 김석환 외 2인 공역, “The C Programming Language”, 2/E, 대영사, 2004.

[7] "C reference", cppreference.com, 2024 of viewing the site, https://en.cppreference.com/w/c.



이 강의자료는 저작권법에 따라 보호받는 저작물이므로 무단 전제와 무단 복제를 금지하며,
내용의 전부 또는 일부를 이용하려면 반드시 저작권자의 서면 동의를 받아야 합니다.

Copyright © Clickseo.com. All rights reserved.