

5강

반복

동양미래대학교 강환수 교수

본 강의 사용 및 참조 자료

▶ Perfect C, 3판, 강환수 외 2인 공저, 인피니티북스, 2021



7장 반복



목차

- 1 반복 while, do while
- 2 반복 for
- 3 break와 continue
- 4 중첩된 반복



01

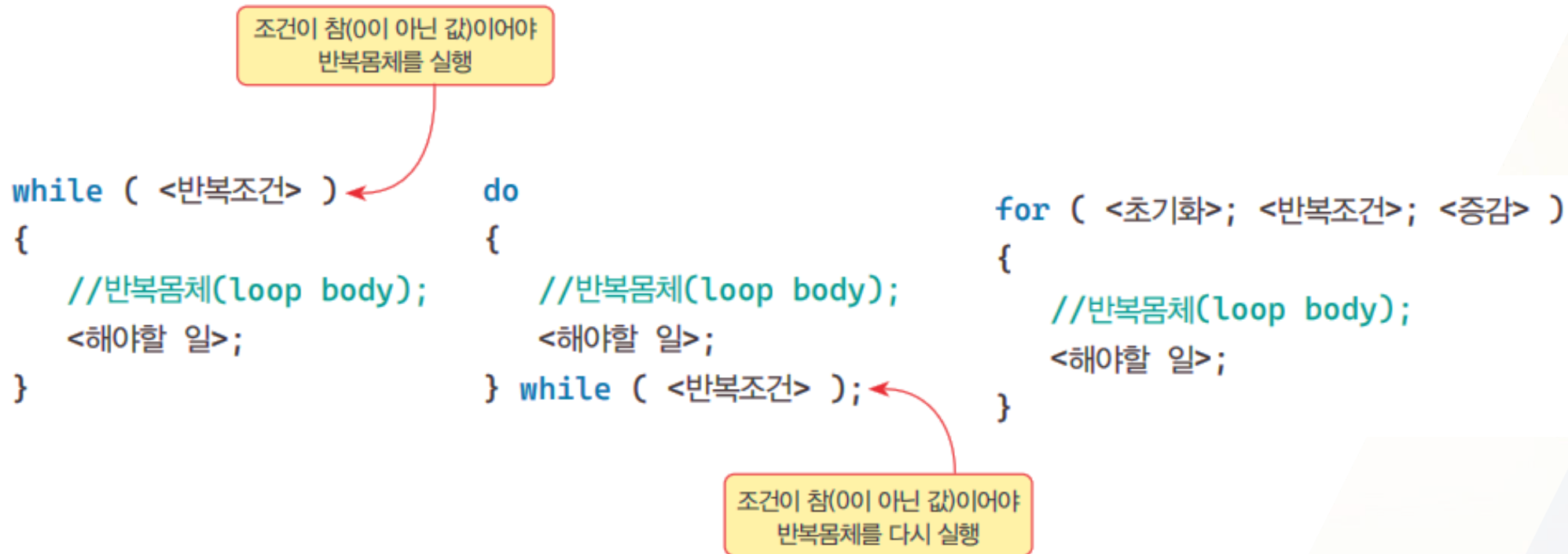
반복 while, do while

➤ 반복

- 순환 또는 루프(loop)라는 표현도 함께 사용
- 반복 몸체(repetition body)
 - 반복 조건을 만족하면 일정하게 반복되는 블록
 - 중괄호 사용 {...}



➤ while, do while, for 세 가지 종류의 반복 구문



여러 섭씨 온도와 해당하는 화씨 온도를 출력

반복 while, do while

- ▶ 함수 printf()를 3번 반복 호출
 - 섭씨 온도는 12.46도에서 10씩 2번 증가
 - 각각의 화씨 온도를 출력

| 섭씨(C) | 화씨(F) |
|-------|-------|
| ----- | |
| 12.46 | 54.43 |
| 22.46 | 72.43 |
| 32.46 | 90.43 |



여러 섭씨 온도를 화씨 온도로 바꾸는 반복

반복 while, do while

$$F(\text{화씨온도}) = \frac{9}{5} C(\text{섭씨온도}) + 32$$

9/5로 쓰면 이 결과가 10이므로 부정확한 결과가 나온다.

```
double celsius = 12.46;

printf("%8.2lf %8.2lf\n", celsius, 9.0 / 5 * celsius + 32);
celsius += 10;
printf("%8.2lf %8.2lf\n", celsius, 9.0 / 5 * celsius + 32);
celsius += 10;
printf("%8.2lf %8.2lf\n", celsius, 9.0 / 5 * celsius + 32);
```

| 섭씨(C) | 화씨(F) |
|-------|-------|
| 12.46 | 54.43 |
| 22.46 | 72.43 |
| 32.46 | 90.43 |

실습예제

반복 while, do while

Prj01

01cel2far3.c

3개의 섭씨 온도를 화씨 온도로 변환

난이도: ★

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     double celsius = 12.46;
06
07     printf(" 섭씨(C) 화씨(F)\n");
08     printf("%8.2lf %8.2lf\n", celsius, 9.0 / 5 * celsius + 32);
09     celsius += 10;
10     printf("%8.2lf %8.2lf\n", celsius, 9.0 / 5 * celsius + 32);
11     celsius += 10;
12     printf("%8.2lf %8.2lf\n", celsius, 9.0 / 5 * celsius + 32);
13
14     return 0;
15 }
```

섭씨 온도 celsius와 변환식으로 화씨 온도 출력, 변환식에서 9.0으로 해야 나누기 연산의 결과가 실수로 나오니 주의를 요함

| 섭씨(C) | 화씨(F) |
|-------|-------|
| 12.46 | 54.43 |
| 22.46 | 72.43 |
| 32.46 | 90.43 |



do while 문 개요

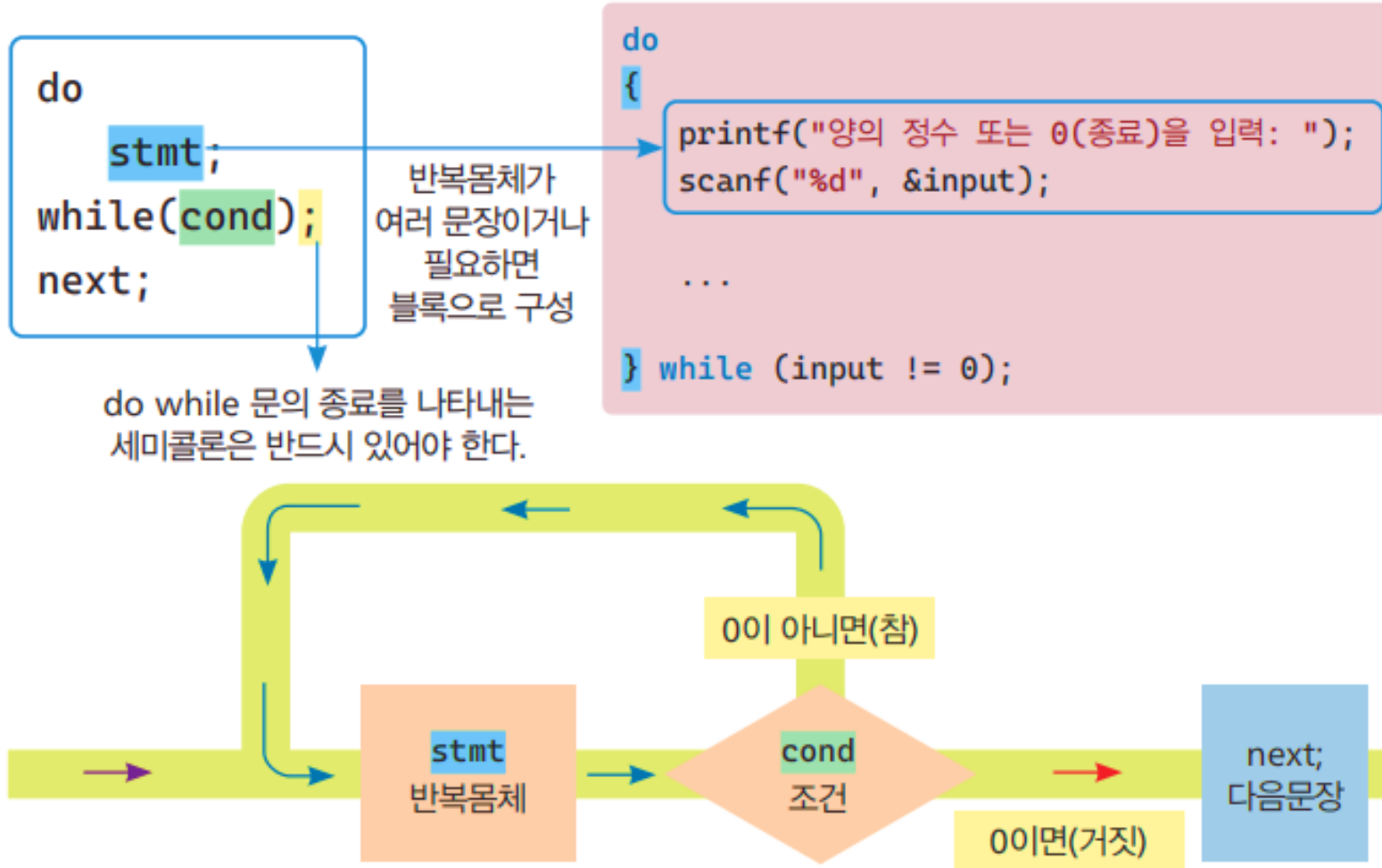
반복 while, do while

- do while 문은 반복몸체 수행 후에 반복 조건을 검사
 - 반복 조건을 나중에 검사해야 하는 반복에 적합
 - while 문은 반복 전에 반복 조건을 평가



do while 문 구조와 제어흐름

반복 while, do while



센티널 값 검사에 유용

반복 while, do while

- 센티널 값(sentinel value)
 - 반복의 종료를 알리는 특정한 자료 값
- 입력 후에 반복 검사를 진행하는 처리 과정
 - do while 문으로 구현이 적합



실습예제 1/2

반복 while, do while

Prj05 05dowhile.c 메뉴 주문 반복

난이도: ★

```
01 #define _CRT_SECURE_NO_WARNINGS
02 #include <stdio.h>
03
04 int main(void)
05 {
06     int input;
07     do
08     {
09         printf("[0]종료 [1]아메리카노 [2]카페라떼 [3]카푸치노 \n");
10         printf("주문할 커피 또는 종료(0)를 입력 >> ");
11         scanf("%d", &input);
12     } while (input != 0); //while (input);
13
14     return 0;
15 }
```

조건식 (input != 0)을 사용하므로 0이 아니어야
9번 줄로 이동하여 반복하며, 0이면 반복을 종료
하고, 조건식 (input != 0)은 (input)과 같음



실습예제 2/2

반복 while, do while

[0]종료 [1]아메리카노 [2]카페라떼 [3]카푸치노
주문할 커피 또는 종료(0)를 입력 >> 2

[0]종료 [1]아메리카노 [2]카페라떼 [3]카푸치노
주문할 커피 또는 종료(0)를 입력 >> 3

[0]종료 [1]아메리카노 [2]카페라떼 [3]카푸치노
주문할 커피 또는 종료(0)를 입력 >> 0



02

반복 for

▶ 반복문 for

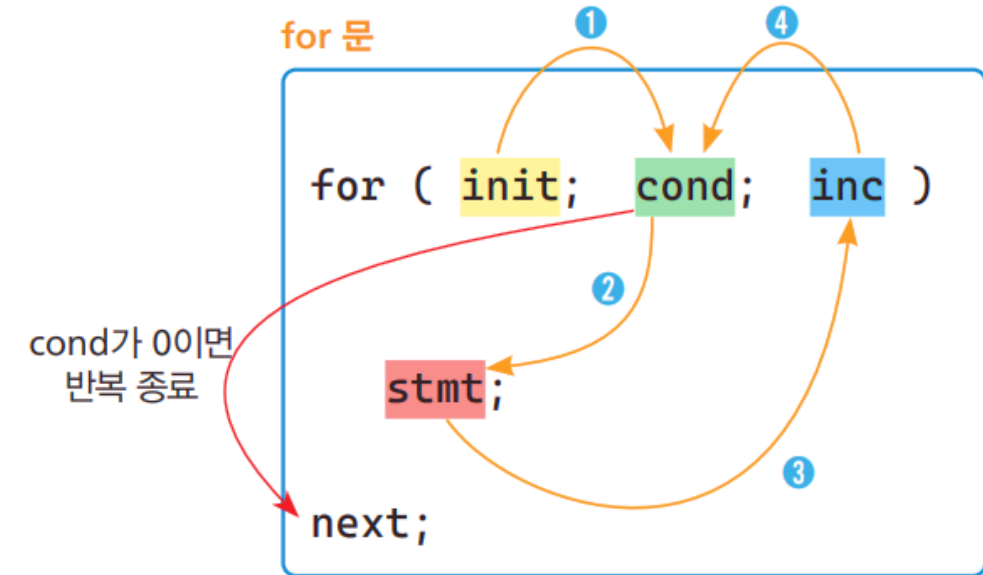
■ for (init; cond; inc) stmt;

- init: 주로 초기화(initialization)
 - ▶ 변수 선언에 의한 초기화도 가능
 - ▶ 여기서 선언된 변수는 반복 for 문 내에서만 사용 가능
- cond: 반복 조건을 검사(condition)
- inc: 주로 반복을 결정하는 제어 변수의 증감(increment)을 수행



for 문 제어 흐름

반복 for



```
for (i=1; i<=10; i++)  
    printf("%3d", i);
```

반복문체가
여러 문장이거나
필요하면
블록으로 구성

```
for (i=1; i<=10; i++)  
{  
    printf("%3d", i);  
    ...  
}
```

for 구문으로 일정 횟수 반복

▶ 괄호 내부, 2개의 세미콜론은 반드시 필요

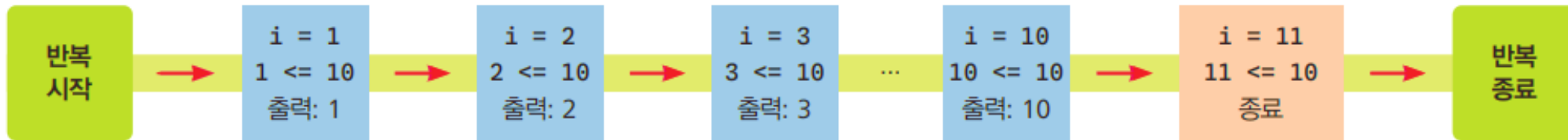
- 반복조건 cond를 아예 제거하면 반복은 무한히 계속

초기화 문장은 단 한번만 실행된다.

```
for (int i = 1; i <= 10; i++)
    printf("%3d:", i);
```

이 부분은 ++i, i = i+1, i += 1 모두 가능하다.

```
int i = 0;
while (i <= 10)
{
    printf("%3d", i);
    i++;
}
```



실습예제

반복 for

Prj06

06forbasic.c

for 구문으로 일정 횟수 반복

난이도: ★

```
01 #include <stdio.h>
02 #define MAX 5
03
04 int main(void)
05 {
06     int i;
07     for (i = 1; i <= MAX; i++)
08         printf("반복 %d\n", i);
09
10     printf("\nfor 종료 이후 i => %d\n", i);
11
12     return 0;
13 }
```

조건식 $i \leq \text{MAX}$ 는 전처리 수행 후, MAX가 5로 대체되어 $i \leq 5$ 가 되며, i 가 5보다 큰 6인 경우 조건식이 거짓이 되어 반복을 종료

증감의 $i++$ 는 반복문체인 8번 줄의 문장이 실행된 이후 실행

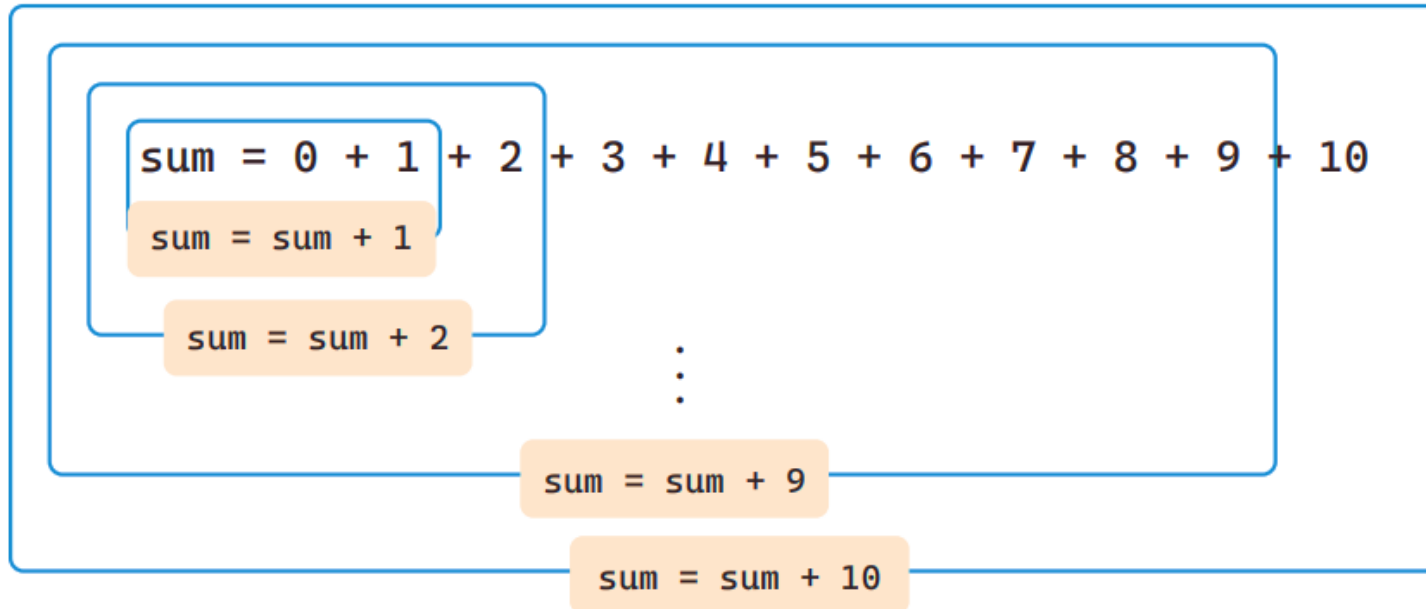
반복 1
반복 2
반복 3
반복 4
반복 5

for 종료 이후 $i \Rightarrow 6$



for 문으로 합 구하기 1/2

- ▶ for 문을 이용하여 1에서 10까지 합을 구하는 모듈
 - 제어 변수 i 를 이용
 - 1부터 10까지 순회
 - 순회하는 제어 변수 i 값을 계속 합하여 변수 sum 에 누적



for 문으로 합 구하기 2/2

반복 for

➤ for 문을 이용하여 1에서 10까지 합을 구하는 다양한 방법

이 부분은 $i = i+1$, $++i$, $i+=1$ 모두 가능하다.

```
for (i=1, sum=0; i<=10; i++)  
    sum += i;
```

```
for (i=1, sum=0; i<=10; sum += i++);
```

```
for (i=1, sum=0; i<=10; )  
    sum += i++;
```

```
for (i=0, sum=0; i<=9; sum += ++i);
```

```
for (i=0, sum=0; i<=9; )  
    sum += ++i;
```

반복문체에 있는 문장을 그대로 증감
부분에 배치해도 같은 기능을 수행한다.



for 문과 while 문의 비교

▶ for 문

- 주로 반복 횟수를 제어하는 제어 변수를 사용
- 초기화와 증감 부분이 있는 반복문에 적합

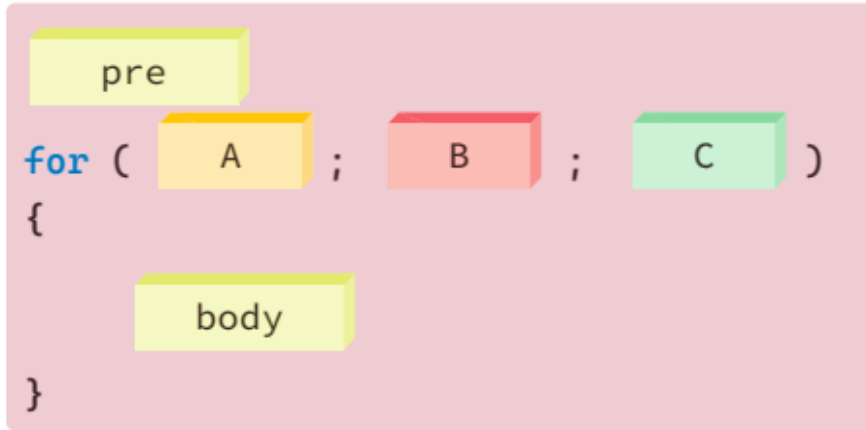
▶ while 문

- 구조가 간단하므로 다양한 구문에 이용
- 반복 횟수가 정해지지 않고
특정한 조건에 따라 반복을 결정하는 구문에 적합
- for 문과 while 문은 서로 변환이 가능

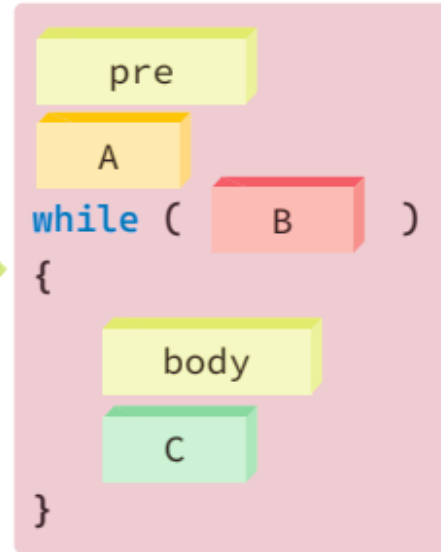


for 문과 while 문의 비교

반복 for



↔ 변환가능



```
sum = 0;
for (i = 1; i <= MAX; i++)
{
    sum += i;
}
```

↔ 변환가능

```
sum = 0;
i = 1;
while (i <= MAX)
{
    sum += i;
    i++;
}
```

sum += i++; 로 가능



03

break와 continue

➤ break 문장

- 반복 내부에서 반복을 종료

```
for ( ; ; )  
{  
    ...  
    break;  
    ...  
}  
next;
```

```
while ( ... )  
{  
    ...  
    break;  
    ...  
}  
next;
```

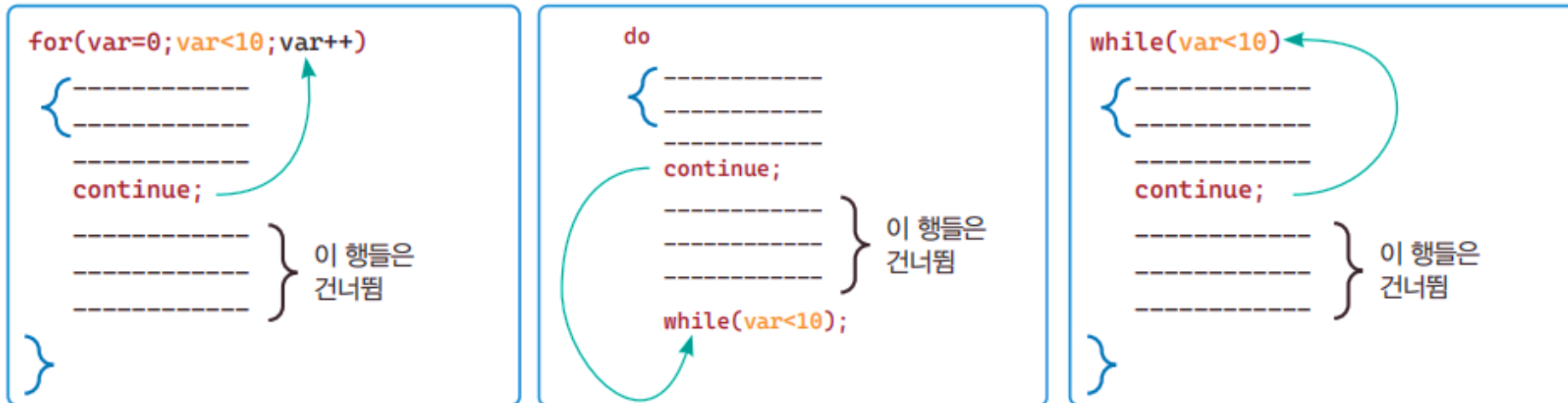
```
do  
{  
    ...  
    break;  
    ...  
} while ( ... );  
next;
```



반복의 계속 continue

break와 continue

- continue는 자신이 속한 가장 근접한 반복에서 다음 반복을 실행



실습예제

break와 continue

Prj12

12continue.c

3으로 나누어지지 않는 정수 출력

난이도: ★

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     const int MAX = 15;
06
07     printf("1에서 %d까지 정수 중에서 3으로 나누어 떨어지지 않는 수\n", MAX);
08     for (int i = 1; i <= MAX; i++)
09     {
10         if (i % 3 == 0) // (!(i % 3))
11             continue;
12         printf("%3d", i);
13     }
14     puts("");
15
16     return 0;
17 }
```

continue를 만나면 실행되지 않고 다음 반복을 위해 i++로 이동

조건식 (i % 3 == 0)은 3으로 나누어 떨어지면 참, 떨어지지 않으면 거짓으로, (!(i % 3))으로도 가능

continue를 만나지 않으면 이 출력문이 실행

1에서 15까지 정수 중에서 3으로 나누어 떨어지지 않는 수

1 2 4 5 7 8 10 11 13 14



무한 반복

break와 continue

무한반복

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

무한반복

```
for ( ; 1 ; )  
{  
    ...  
}
```

무한반복

```
while ( 1 )  
{  
    ...  
}
```

무한반복

```
do  
{  
    ...  
} while ( 1 )
```

오류

```
while ( )  
{  
    ...  
}
```

오류

```
do  
{  
    ...  
} while ( )
```



04

중첩된 반복

- for 문 내부에 for 문이 존재
- 제어 변수는 m, n
 - 외부 for 문의 제어 변수는 m 이며,
내부 for 문의 제어 변수는 n
 - 외부 반복에서 m 은 1에서 4까지 반복
 - 각각의 m 에 대해, 내부 반복에서 n 이 1에서 3까지 반복



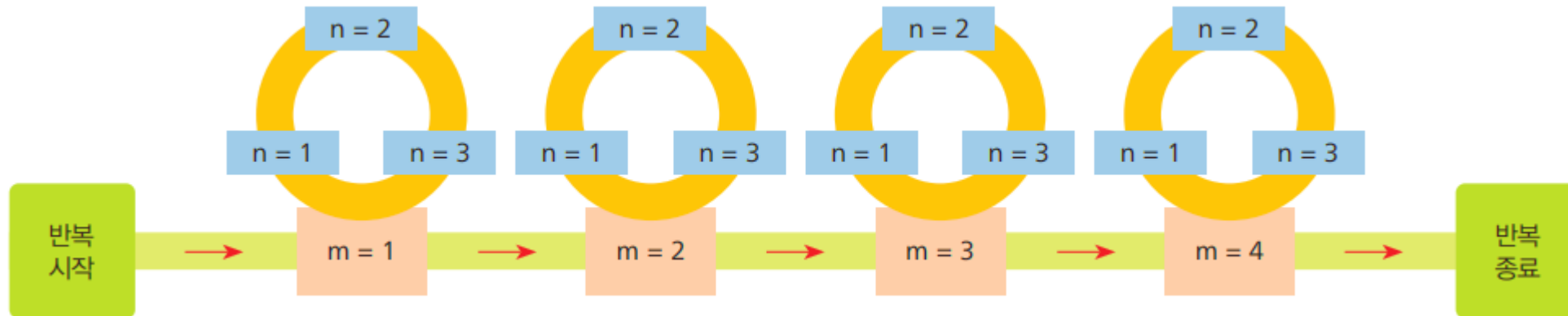
중첩된 for 문 2/2

중첩된 반복

```
for (m = 1; m <= 4; m++)  
{  
  내부 반복 { for (n = 1; n <= 3; n++)  
              ...  
            }  
}
```

외부 반복은 제어변수 m으로 1에서 4까지 반복한다.

내부 반복은 외부 반복의 한 횟수마다 반복되는 부분으로
제어변수 n으로 1에서 3까지 반복한다.



실습예제 1/3

- ▶ 내부 반복과 외부 반복에서 각각의 변수 값의 변화를 이해
 - 외부 반복에서 1에서 5까지 반복
 - 내부 반복에서 1에서 7까지 반복
 - 각각의 변수 값을 출력



실습예제 2/3

중첩된 반복

Prj15

15nestedloop.c

내부 반복과 외부 반복에서 각각의 변수 값의 변화를 이해

난이도: ★

```
01 #include <stdio.h>
02
03 int main(void)
04 {
05     int m, n;
06     for (m = 1; m <= 5; m++)
07     {
08         printf("m = %-2d\n", m);
09         for (n = 1; n <= 7; n++)
10             printf("n = %-3d", n);
11         puts("");
12     }
13
14     return 0;
15 }
```

외부 반복의 for 문으로 1에서 5까지 반복

내부 반복의 for 문으로 1에서 7까지 반복



실습예제 3/3

중첩된 반복

$m = 1$

$n = 1$ $n = 2$ $n = 3$ $n = 4$ $n = 5$ $n = 6$ $n = 7$

$m = 2$

$n = 1$ $n = 2$ $n = 3$ $n = 4$ $n = 5$ $n = 6$ $n = 7$

$m = 3$

$n = 1$ $n = 2$ $n = 3$ $n = 4$ $n = 5$ $n = 6$ $n = 7$

$m = 4$

$n = 1$ $n = 2$ $n = 3$ $n = 4$ $n = 5$ $n = 6$ $n = 7$

$m = 5$

$n = 1$ $n = 2$ $n = 3$ $n = 4$ $n = 5$ $n = 6$ $n = 7$



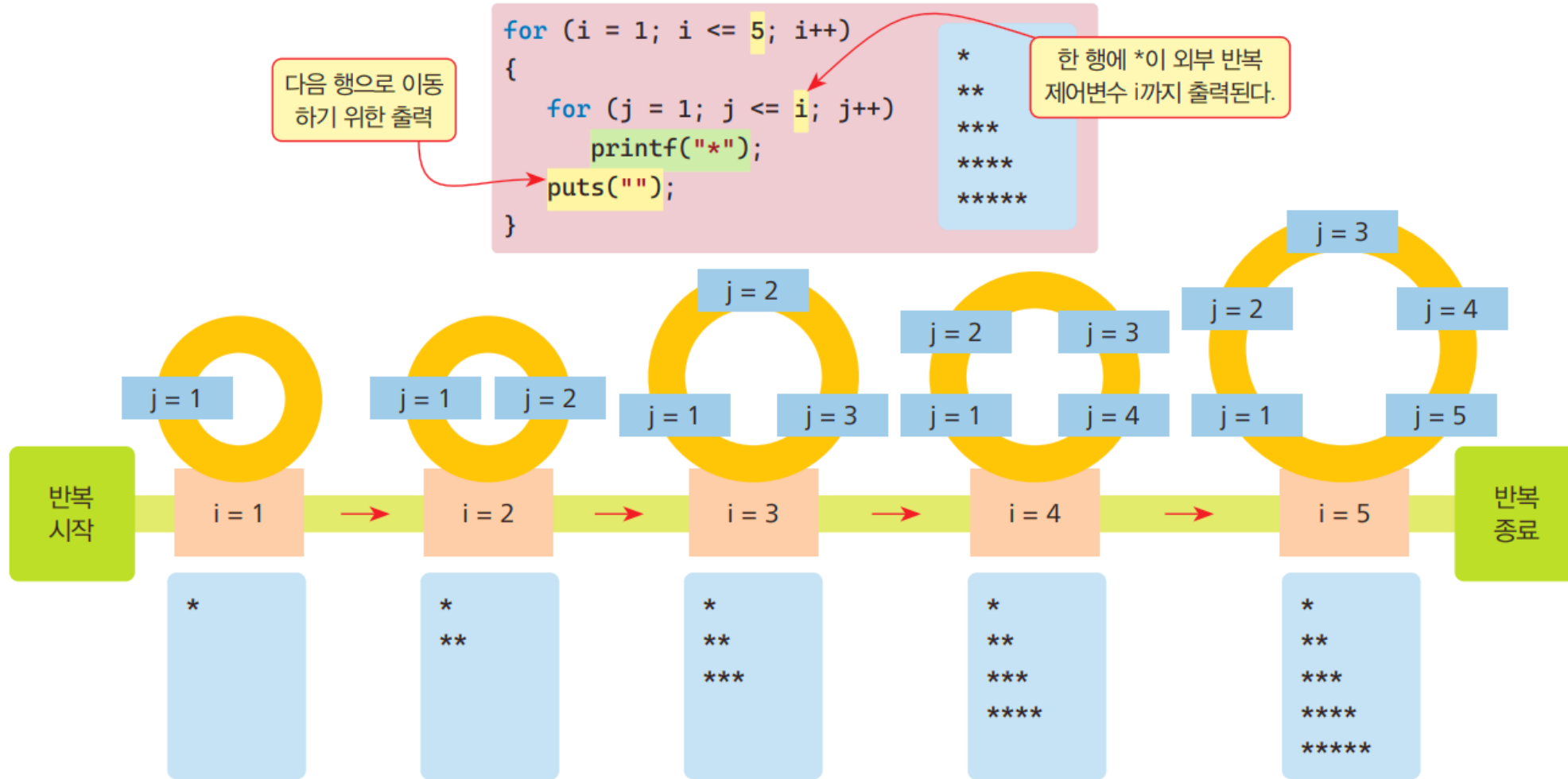
내부 반복이 외부 반복에 의존 1/2

➤ 외부 반복에서 변수 i

- 1 에서 5까지 반복
- 내부 반복에서 제어 변수 j
 - 1에서 외부 반복의 제어 변수 i 까지 반복



내부 반복이 외부 반복에 의존 2/2



구구단 출력 프로그램

=== 구구단 출력 ===

2단 출력

$2*2 = 4$ $2*3 = 6$ $2*4 = 8$ $2*5 = 10$ $2*6 = 12$ $2*7 = 14$ $2*8 = 16$ $2*9 = 18$

3단 출력

$3*2 = 6$ $3*3 = 9$ $3*4 = 12$ $3*5 = 15$ $3*6 = 18$ $3*7 = 21$ $3*8 = 24$ $3*9 = 27$

4단 출력

$4*2 = 8$ $4*3 = 12$ $4*4 = 16$ $4*5 = 20$ $4*6 = 24$ $4*7 = 28$ $4*8 = 32$ $4*9 = 36$

5단 출력

$5*2 = 10$ $5*3 = 15$ $5*4 = 20$ $5*5 = 25$ $5*6 = 30$ $5*7 = 35$ $5*8 = 40$ $5*9 = 45$

6단 출력

$6*2 = 12$ $6*3 = 18$ $6*4 = 24$ $6*5 = 30$ $6*6 = 36$ $6*7 = 42$ $6*8 = 48$ $6*9 = 54$

7단 출력

$7*2 = 14$ $7*3 = 21$ $7*4 = 28$ $7*5 = 35$ $7*6 = 42$ $7*7 = 49$ $7*8 = 56$ $7*9 = 63$

8단 출력

$8*2 = 16$ $8*3 = 24$ $8*4 = 32$ $8*5 = 40$ $8*6 = 48$ $8*7 = 56$ $8*8 = 64$ $8*9 = 72$

9단 출력

$9*2 = 18$ $9*3 = 27$ $9*4 = 36$ $9*5 = 45$ $9*6 = 54$ $9*7 = 63$ $9*8 = 72$ $9*9 = 81$



실습예제 2/2

중첩된 반복

lab4mtable.c

난이도: ★

```
01  #include <stdio.h>
02  #define MAX 9
03
04  int main(void)
05  {
06      for (int i = 2; i <= MAX; i++)
07      {
08          printf("%5d단 출력\n", i);
09          for (int j = 2; j <= MAX; j++)
10              printf();
11          printf();
12      }
13
14      return 0;
15  }
```

```
10      printf("%d*d = %2d ", i, j, i * j);
11      printf("\n");
```



정 리 하 기



정리하기

- 반복 구문의 필요성을 이해하고 반복문을 사용한다.
- C 언어에서 제공하는 반복 구문
for, while, do while 구문을 적절히 사용한다.
- 조건 검사를 처음에 하는 반복은 while 구문이 적합하다.
- 조건 검사를 나중에 하는 반복은 do while 구문이 적합하다.
- 일정한 규칙으로 제어변수에 증감이 있는 경우는 for 문이 적합하다.
- 반복문에서 제어변수와 센티널 값을 적절히 사용한다.
- 반복 내부에서 break와 continue 문장을 적절히 사용한다.
- 반복 내부에 다시 반복이 있는 중첩된 반복을 사용한다.

6강

다음시간안내

배열