

# 파이썬 프로그래밍 강의 노트 #07

---

## 반복문

# 반복문을 사용하지 않는 입력과 합계

- ▣ 정수 세 개를 입력받고 합을 구해서 화면에 출력하는 프로그램 작성

```
# 방법 1
n1 = int(input("정수 한 개를 입력하세요: "))
n2 = int(input("정수 한 개를 입력하세요: "))
n3 = int(input("정수 한 개를 입력하세요: "))
sum = n1 + n2 + n3
print(sum)
```

# 반복문을 사용하지 않는 입력과 합계

```
# 방법 2
sum = 0
n1 = int(input("정수 한 개를 입력하세요: "))
sum += n1
n2 = int(input("정수 한 개를 입력하세요: "))
sum += n2
n3 = int(input("정수 한 개를 입력하세요: "))
sum += n3
print(sum)
```

# 반복문을 사용하지 않는 입력과 합계

```
# 방법 3
sum = 0
n = int(input("정수 한 개를 입력하세요: ")) # (1)
sum += n                                     # (2)
n = int(input("정수 한 개를 입력하세요: "))
sum += n
n = int(input("정수 한 개를 입력하세요: "))
sum += n
print(sum)
```

# 반복문을 사용하지 않는 입력과 합계

```
# 방법 4
def getSum(s):
    n = int(input("정수 한 개를 입력하세요: "))
    s += n
    return s

sum = 0
sum = getSum(sum)
sum = getSum(sum)
sum = getSum(sum)
print(sum)
```

# 반복문 (Loop Statements)

- 프로그램을 작성하고 실행시키는 이유 중 한 가지는 사람이 하기 싫은 단순하고 반복적인 작업을 시키기 위해서임
- 반복문을 사용하려면 종료 조건을 이해해야 함
  - 종료조건이란 반복을 종료시키는 조건
  - 주로 True 또는 False로 결과가 나타나는 조건식이 많이 사용됨
- 반복문이 필요한 예
  - 버스 정류장에서 7016 버스가 있으면 탄다

```
if "7016버스가 정류장에 있으면":  
    버스타기()
```

# 반복문 (Loop Statements)

## □ 만약 버스가 없으면 기다린다

```
if "7016버스가 정류장에 있으면":  
    버스타기()  
if "7016버스가 정류장에 없으면":  
    기다리기()
```

## □ 버스를 타려면, 두 과정을 반복해야 함

```
if "7016버스가 정류장에 있으면":  
    버스타기()  
if "7016버스가 정류장에 없으면":  
    기다리기()  
if "7016버스가 정류장에 있으면":  
    버스타기()  
if "7016버스가 정류장에 없으면":  
    기다리기()  
... # 생략 표시는 두 개의 if문이 반복됨을 보임
```

# 반복문 (Loop Statements)

## □ 종료 조건이 만족될 때까지 반복하라

반복 조건이 만족될 때까지:

기다림

```
if "7016버스가 정류장에 있으면":  
    버스타기()
```

## □ 종료 조건

### ■ 버스가 오면

```
while "7016 버스가 정류장에 없으면":
```

기다림

```
if "7016버스가 정류장에 있으면":  
    버스타기()
```



# while 문

while 조건표현식:  
코드 블록

## □ while 문

- if문은 조건표현식이 만족되면 한 번만 실행되지만, while 문은 조건이 만족되지 않을 때까지 계속 반복됨 (조건 반복형)
- 조건표현식의 결과가 False가 되면(혹은 0, 빈 문자열 등) 코드블록이 실행되지 않고 다음 실행 구문으로 이동
- 조건에 따른 반복 작업에 최적화 되어 있음

# while 문

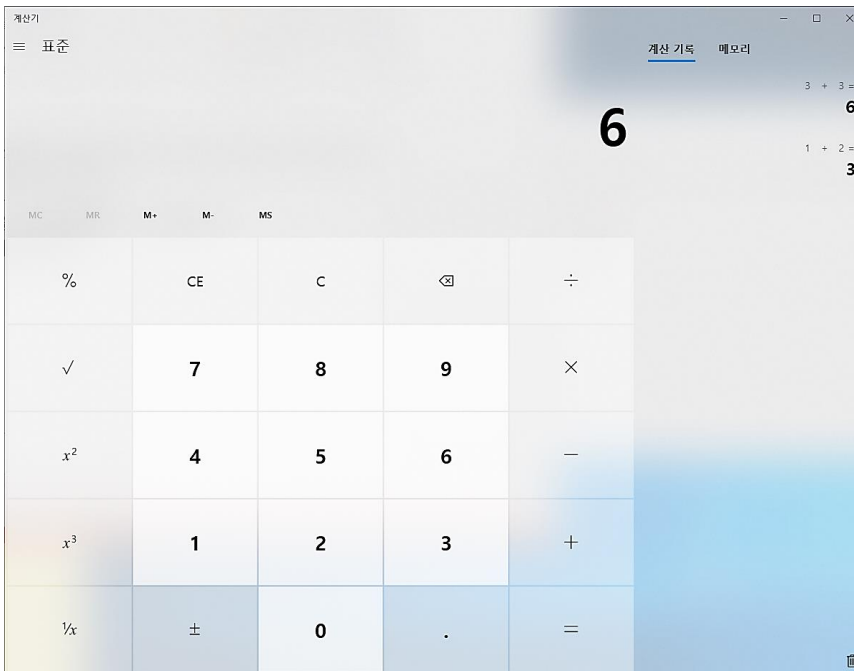
## □ while문의 예

```
>>> n = 1
>>> while n < 4:
...     print(n)
...     n += 1
...
1
2
3
```

# while 문

## □ 1~10까지의 합계 구하기

- $1 + 2 + 3 + \dots + 9 + 10$
- 계산 방법을 다른 사람에게 설명한다면?
- 계산기를 사용한다면?



# while 문

## ▣ 코드로 표현하면?

```
sum = 0
sum = 0 + 1
sum = 0 + 1 + 2
sum = 0 + 1 + 2 + 3
...
sum = 0 + 1 + 2 + 3 + ... + 9
sum = 0 + 1 + 2 + 3 + ... + 9 + 10
```

```
sum = 0
sum += 1
sum += 2
sum += 3
...
sum += 9
sum += 10
```

# while 문

```
sum = 0
n = 1    # 1
sum += n
n += 1   # 2
sum += n
n += 1   # 3
...
n += 1   # 10
sum += n
print(sum)
```

n <= 10

```
sum = 0
n = 1
if n <= 10:
    sum += n
    n += 1
if n <= 10:
    sum += n
    n += 1
...
print(sum)
```

n <= 10

# while문

## □ 반복문으로 작성

```
sum = 0
n = 1
while n <= 10:
    sum += n
    n += 1
print(sum)
```

# 실습문제 1

## □ 문제

- 사용자로부터 숫자를 입력받아 합이 100을 넘을 때까지 반복하여 더하는 프로그램을 작성하세요. 합이 100을 넘으면 프로그램을 종료하고 최종 합을 출력하세요.

## ■ 요구사항

- 사용자가 입력하는 숫자는 모두 0보다 큰 양수로 가정
- While 사용

```
숫자를 입력하세요: 5
숫자를 입력하세요: 4
숫자를 입력하세요: 5
숫자를 입력하세요: 5
숫자를 입력하세요: 6
숫자를 입력하세요: 5
숫자를 입력하세요: 5
```

# 실습문제 1

```
def sum_until_100():  
    total = 0  
    while total <= 100:  
        num = int(input("숫자를 입력하세요: "))  
        total += num  
    print(f"최종 합계: {total}")  
  
# 함수 호출  
sum_until_100()
```



# 실습문제 2

---

## □ 문제

- 사용자가 "비밀번호"를 입력할 때까지 반복하여 입력을 받는 프로그램을 작성하세요. 사용자가 올바른 비밀번호를 입력하면 "인증 성공"을 출력하고 프로그램을 종료하세요.
- 요구사항
  - 비밀번호는 "ROKEY"로 가정합니다.
  - While 사용

## 실습문제 2

```
def user_authentication():  
    password = "rokey"  
    while True:  
        user_input = input("비밀번호를 입력하세요: ")  
        if user_input == password:  
            print("인증 성공")  
            break  
        else:  
            print("비밀번호가 틀렸습니다. 다시 시도하세요.")  
  
# 함수 호출  
user_authentication()
```

# for 문

- for문은 주로 정해진 횟수만큼 반복하거나 제한된 개수의 요소들을 순회하면서 처리할 때 주로 사용하는 반복문

- 자주 활용되는 반복문

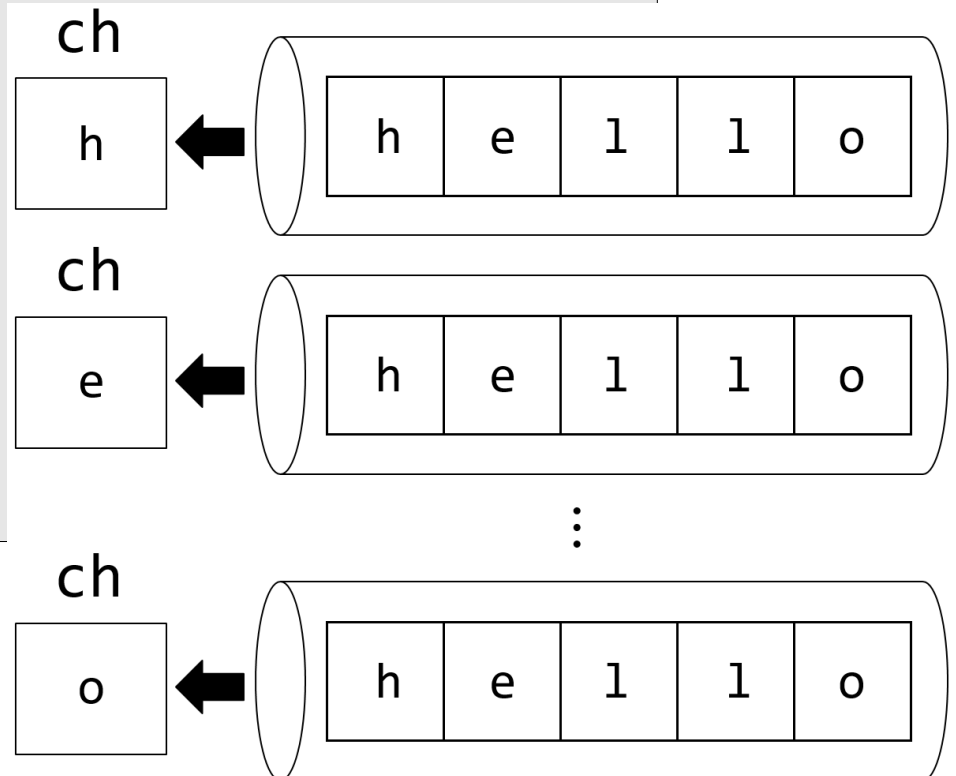
```
for 변수 in 순서가_있는_객체:  
    코드_블록
```

- 순서가 있는 객체
  - 문자열, range 객체, 리스트(list), 튜플(tuple) 등
- 순서가\_있는\_객체의 각 요소들은 변수에 치환되고 해당 변수를 이용해서 코드\_블록을 실행함
- 반복 횟수는 순서가\_있는\_객체의 크기 혹은 아이템의 개수

# for 문

- for문과 문자열을 이용해서 글자들을 한 줄에 한 개씩 화면에 출력

```
>>> s = "hello"
>>> for ch in s: # for ch in "hello":
...     print(ch)
...
h
e
l
l
o
```



# for 문과 range()

## □ for 반복문

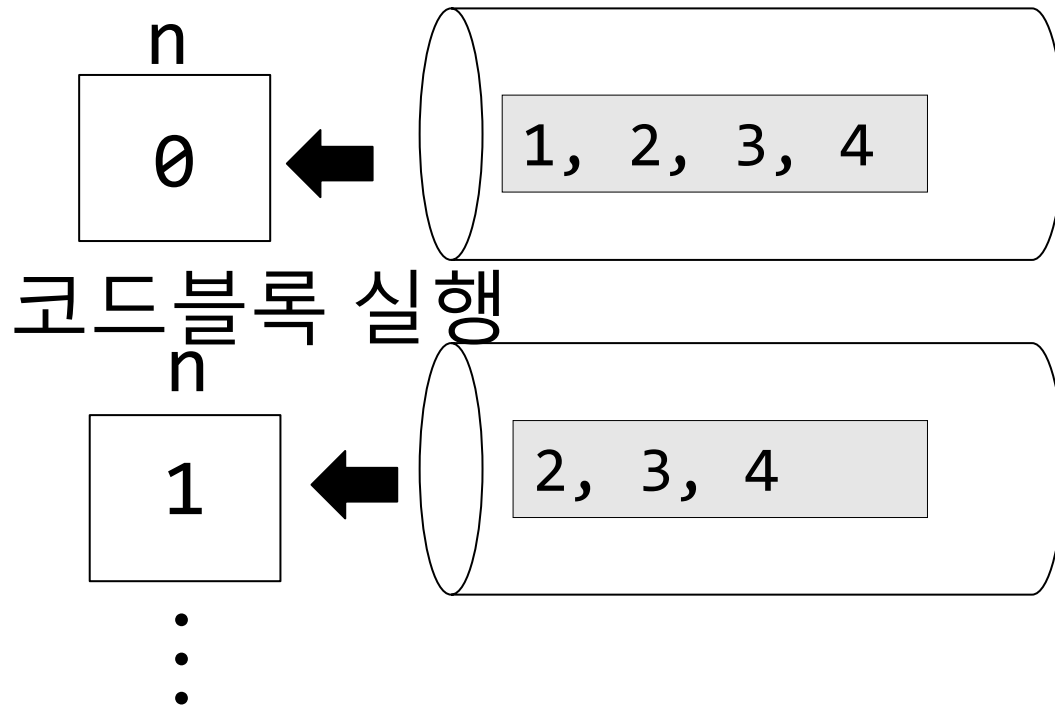
- 일정 정수 범위 또는 패턴에 대해서 수행되는 경우가 많은데 (예: 특정 횟수만큼 혹은 특정 주기를 반복), 이럴 때에는 range 함수를 이용

```
for 변수 in range(범위):  
    코드블록
```

- range()는 range 객체를 반환

# for 문과 range()

```
for n in range(5):  
    코드블록
```



# for 문과 range()

- range()의 범위는 다양하게 정의 가능함

```
range(c)      # range(0, 1, 2, ..., c - 1)
range(c, d)    # range(c, c + 1, c + 2, ..., d - 1)
range(c, d, e) # range(c, c + e, c + 2 * e,
                # ..., c + n * e (c < d and e >
                # 0이면, c + n * e는 d보다 작은 가장 큰 정수, c > d
                # and e < 0이면, c + n * e는 d보다 큰 가장 작은
                # 정수))
```

- 예)

```
a = range(-5, -2)      # a = range(-5, -4, -3)
a = range(-5, 3, 2)     # a = range(-5, -3, -1, 1)
a = range(7, 1, -3)     # a = range(7, 4)
a = range(7, -3, -2)    # a = range(7, 5, 3, 1, -1)
```

# for 문과 range()

## ▣ 횟수를 세는 range 객체

```
>>> for i in range(3):  
...     print("hello")  
...  
hello  
hello  
hello
```



## for 문과 range()

### ▣ 1~10까지의 합 구하기

```
>>> sum = 0
>>> for n in range(1, 11): # n을 1~10까지 변경
...     sum += n
...
>>> print(sum)
55
```

# 실습문제 3

---

## □ 문제

- 사용자로부터 문자열을 입력받아, 그 문자열을 역순으로 출력하는 프로그램을 작성하세요.
- 요구사항
- For 활용

# 실습문제 3

## ▣ 최종 코드

```
def reverse_string(s):  
    reversed_s = ""  
    for char in s:  
        reversed_s = char + reversed_s  
    return reversed_s  
  
# 사용자 입력  
s = input("문자열을 입력하세요: ")  
  
# 함수 호출 및 결과 출력  
reversed_s = reverse_string(s)  
print(f"역순 문자열: {reversed_s}")
```

# 실습문제 4

---

## □ 문제

- 사용자로부터 정수  $N$ 을 입력받아,  $N$ 개의 랜덤 숫자를 생성하고 그 숫자들의 합을 계산하는 프로그램을 작성하세요.

## □ 요구사항

- 랜덤 숫자는 1부터 100 사이의 정수로 생성됩니다.
- For 활용
- `import random`

# 실습문제 4

## □ 최종 코드

```
import random
def sum_of_random_numbers(n):
    total = 0
    for _ in range(n):
        rand_num = random.randint(1, 100)
        print(f"생성된 랜덤 숫자: {rand_num}")
        total += rand_num
    return total

# 사용자 입력
n = int(input("랜덤 숫자의 개수를 입력하세요: "))

# 함수 호출 및 결과 출력
total_sum = sum_of_random_numbers(n)
print(f"랜덤 숫자들의 합: {total_sum}")
```

# 중첩 반복문

## □ 중첩 반복문

- 반복문의 코드블록에 다른 반복문이 있는 경우

```
for 변수1 in 순서가_있는_객체1:
    for 변수2 in 순서가_있는_객체2:
        코드_블록
while 조건식1:
    while 조건식2:
        코드_블록:
for 변수1 in 순서가_있는_객체1:
    while 조건식1:
        코드_블록
while 조건식1:
    for 변수1 in 순서가_있는_객체1:
        코드_블록
```

# 실습문제 5

## □ 문제

- 8x8 체스판을 출력하는 프로그램을 작성하세요. 체스판은 '#'과 ' ' (공백)로 구성되며, 다음과 같이 출력되어야 합니다:

```
# # # # # # # #  
# # # # # # # #  
# # # # # # # #  
# # # # # # # #  
# # # # # # # #  
# # # # # # # #  
# # # # # # # #  
# # # # # # # #
```

## □ 요구사항

- 중첩 반복문 for 사용

## 실습문제 5

### ▣ 최종 코드

```
def print_chess_board():  
    for i in range(8):  
        for j in range(8):  
            if (i + j) % 2 == 0:  
                print("#", end=' ')  
            else:  
                print(" ", end=' ')  
        print()  
  
# 함수 호출  
print_chess_board()
```



# break 문

## □ break문

- 반복문(while 또는 for문)의 실행을 중단하고 반복문을 빠져나옴

while 조건식:

코드\_블록\_1

break

코드\_블록\_2



다음\_실행\_코드

# break 문

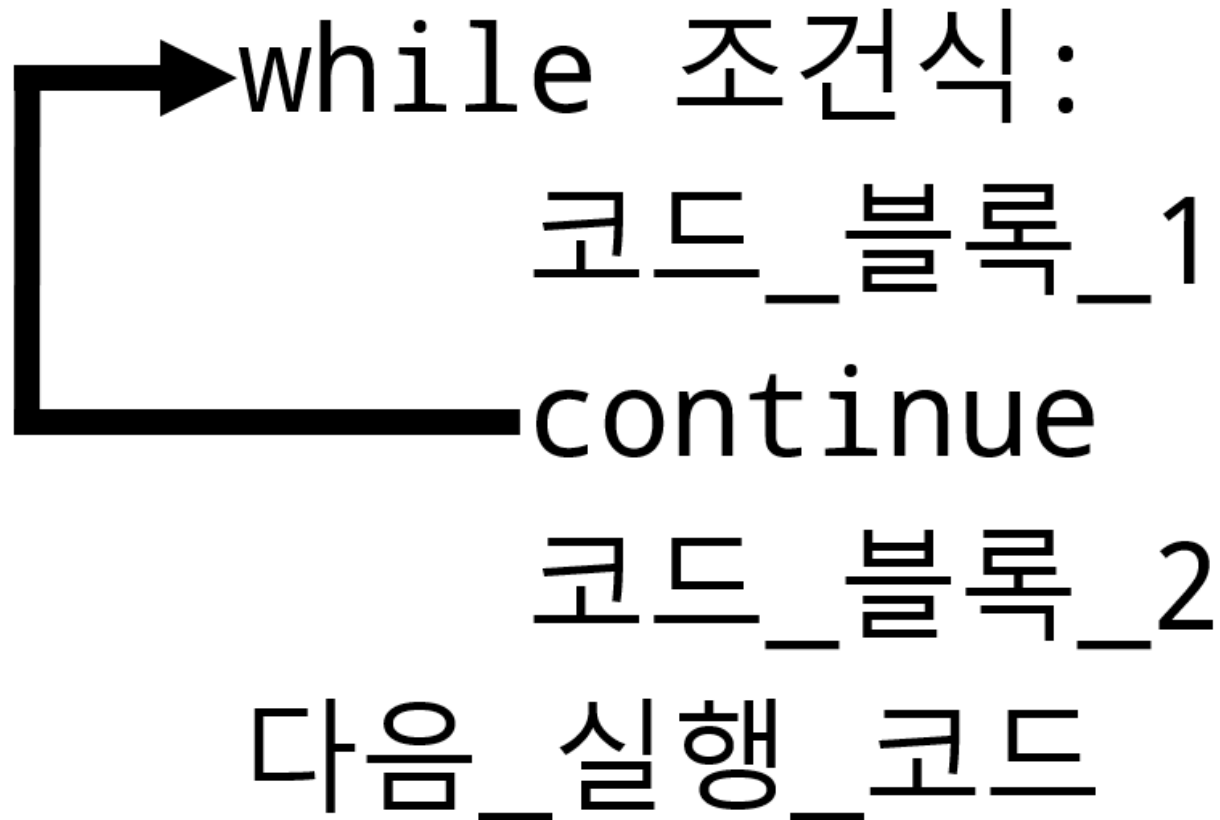
## □ 예제 코드

- 문자열 s에서 첫 번째 숫자의 위치를 출력하는 프로그램

```
s = "what are the 10 best-selling products in this  
shopping mall?"  
idx = 0  
while idx < len(s):  
    if s[idx].isdigit():  
        print(idx)  
        break  
    idx += 1
```

## continue문

- 반복문에서 사용되는 continue문은 남은 코드의 실행을 중단하고 반복문의 처음으로 되돌아감



# continue문

## □ 예제 코드

- 사용자로부터 10개의 양의 정수를 입력받고 합을 구하는 프로그램 작성

```
count = 0
sum = 0
while count < 10:
    n = int(input("양의 정수를 입력하세요: "))
    if n <= 0: # 0 또는 음수이면
        continue # 반복문의 처음으로 이동
    # CODE_BLOCK_1
    sum += n
    count += 1
    print("count =", count)
print("합 =", sum)
```

# 실습문제 6

---

## □ 문제

- 사용자로부터 정수를 입력받아, 1부터 해당 정수까지의 합을 계산하는 프로그램을 작성하세요.

## □ 요구사항

- 정수 > 1
- Input 사용
- For 사용
- 함수 사용

# 실습문제 7

---

## □ 문제

- 사용자로부터 정수를 입력받아, 해당 정수만큼 별 (\*) 을 한 줄에 출력하는 프로그램을 작성하세요.

## □ 요구사항

- Input 사용
- While 사용
- 함수 사용

# 실습문제 8

---

## □ 문제

- 컴퓨터가 1부터 100 사이의 숫자를 랜덤하게 선택하고, 사용자가 그 숫자를 맞추는 게임을 작성하세요. 사용자는 숫자를 입력할 때마다 컴퓨터가 "크다", "작다" 또는 "정답"이라고 응답합니다.

## □ 요구사항

- Import random 사용
- random.randint(1, 100)

# 실습문제 9

## □ 문제

- 사용자로부터 1부터 9 사이의 정수를 입력받아, 해당 숫자의 구구단을 출력하는 프로그램을 작성하세요.

## □ 요구사항

- For 활용

```
1부터 9 사이의 정수를 입력하세요: 5  
5 * 1 = 5  
5 * 2 = 10  
5 * 3 = 15  
5 * 4 = 20  
5 * 5 = 25  
5 * 6 = 30  
5 * 7 = 35  
5 * 8 = 40  
5 * 9 = 45
```



# 실습문제 10

---

## □ 문제

- 사용자로부터 문자열을 입력받아, 문자열 내 모든 숫자의 합을 계산하는 프로그램을 작성하세요.

## □ 요구사항

- isDigit()

# 실습문제 11

---

## □ 문제

- 사용자로부터 정수를 입력받아, 1부터 해당 정수까지의 홀수들만 곱한 결과를 출력하는 프로그램을 작성하세요.

## □ 요구사항

- While 활용

# 실습문제 12

---

## □ 문제

- 사용자로부터 문자열과 특정 문자를 입력받아, 해당 문자열에서 특정 문자가 몇 번 등장하는지, 세는 프로그램을 작성하세요.

## □ 요구사항

- For 활용

# 실습문제 13

---

## □ 문제

- 사용자로부터 두 개의 정수를 입력받아, 첫 번째 정수부터 두 번째 정수까지의 모든 숫자의 합을 계산하는 프로그램을 작성하세요. (첫 번째 정수는 항상 두 번째 정수보다 작다고 가정합니다)

## □ 요구사항

- While 조건 적용

# 실습문제 14

---

## □ 문제

- 구구단을 2단부터 9단까지 출력하는 프로그램을 작성하세요..

## □ 요구사항

- 중첩 for 문 활용

# 실습문제 15

---

## □ 문제

- 사용자로부터 정수를 입력받아, 해당 정수 높이의 별(\*) 피라미드를 출력하는 프로그램을 작성하세요.

## □ 요구사항

- 중첩 for 문 활용

# 실습문제 16

---

## □ 문제

- 사용자로부터 문자열을 입력받아, 문자열 내 각 단어의 각 문자를 한 줄씩 출력하는 프로그램을 작성하세요.

## □ 요구사항

- 중첩 for 문 활용

# 실습문제 17

## □ 문제

- 사용자로부터 정수를 입력받아, 해당 정수 크기의 덧셈 표를 생성하는 프로그램을 작성하세요. 예를 들어, 사용자가 3을 입력하면 다음과 같은 표가 출력되어야 합니다:

```
1 2 3
2 4 6
3 6 9
```

## □ 요구사항

- For 중첩문



# Geek & Geek

