

부스트코스 4주차

📌 Q1. 우리는 1,000이상의 큰 수를 쓸 때, 숫자의 크기를 쉽게 확인할 수 있도록 매 3자리마다 ","를 찍어 숫자를 표기하고 있습니다. 파이썬에도 아래와 같이 쉽게 나타낼 수 있는 편리한 방법이 존재하지요. 하지만! 이번 미션에서는 아래의 기능을 사용하지 않고, 직접 숫자 표기법을 만들어 보도록 합니다. 사용자로부터 1,000이상의 숫자를 입력 받고, 매 3자리마다 ","를 찍어 숫자의 크기를 쉽게 구별할 수 있는 코드를 작성해주세요. [난이도 :

★★/5]

```
# 파이썬으로 쉽게 숫자 표기 작성하는 법
# f"{숫자:,}"
print(f"{1000:,}")
```

✓ 보조 설명 :

이번 문제에서는 파이썬 내장 함수를 사용하지 않고, 직접 숫자 표기법을 만들어보려고 합니다.

구현 순서는 다음과 같습니다 :

1. 사용자로부터 숫자를 입력 받습니다.
2. 입력받은 숫자를 문자열로 변환합니다.
3. 입력된 숫자의 길이를 확인합니다.
4. 숫자의 길이가 3 이하라면, 입력된 숫자를 그대로 출력합니다.
5. 숫자의 길이가 4 이상이라면, 입력된 숫자를 거꾸로 탐색하면서 ","를 매 3자리마다 삽입합니다.
6. 탐색이 끝난 숫자를 다시 뒤집어서 출력합니다.

이렇게 작성된 코드는 입력 예로, 1234567을 받으면 "1,234,567"으로 출력됩니다. ▼출력 예시

```
make_comma(1000000)
1,000,000
```

```
def insert_commas(number):
    number_str = str(number)
```

```

length = len(number_str)

if length <= 3:
    return number_str

reversed_result = []
count = 0

for digit in reversed(number_str):
    if count == 3:
        reversed_result.append(',')
        count = 0
    reversed_result.append(digit)
    count += 1

formatted_number = ''.join(reversed(reversed_result))
return formatted_number

# 사용자로부터 숫자 입력 받기
input_number = int(input("1,000 이상의 숫자를 입력하세요: "))
output = insert_commas(input_number)
print("결과:", output)


```

사용자로부터 숫자 입력 받기

```
input_number = int(input("1,000 이상의 숫자를 입력하세요: "))
```

입력 받은 숫자를 형식에 맞게 포매팅하여 출력

```
formatted_output = insert_commas(input_number)
print("포매팅된 결과:", formatted_output)
```

 **Q2.** 햇빛이 따사로운 어느 날, 문득 읽고 있던 연애소설에서 '사랑'이라는 단어가 몇 개나 등장하는지 궁금해졌습니다. 손으로 하나씩 세기에는 페이지 수가 너무 많아 번거롭게 느껴지는데요, 파이썬이라면 이 문제를 한번에 해결할 수 있다는 생각이 들어 시도해보려고 합니다. 이번주에 배운 내용을 토대로 파이썬을 통해 입력한 문장에서 특정 단어의 갯수를 세어 주는 코드를 작성해봅시다. [난이도 : ★★/5]

- 글은 어떤 글이 좋습니다. 인터넷에서 검색해서 복사 붙여넣기로 변수에 넣어 줍니다.
- 변수에 담긴 글을 함수에 넣어주면 txt 파일로 저장도 함께 되도록 해줍니다.

✓ 보조 설명 :

1. 사용자로부터 문장을 입력 받습니다.
2. 입력받은 문장을 문자열로 저장합니다.
3. 단어의 개수를 저장할 변수를 초기화합니다.

4. 입력된 문장에서 '찾고자 하는 단어(ex. 사랑)'를 찾기 위해 문자열의 메소드를 활용합니다.
5. 메소드를 사용하여 '찾고자 하는 단어'의 개수를 세어 변수에 저장합니다.
6. 단어의 개수를 출력합니다.

▼출력 예시

```
a = """안녕하세요.
반갑습니다. 파이썬 공부는 정말 재밌습니다.
"""

count_word(a, "습니다")
2
```

📌 Q3. 특정 행사에서 방명록을 작성하는 웹 사이트를 개발하려고 합니다. 방명록에는 참석자들의 휴대전화번호가 포함되어야 합니다. 하지만 몇 명의 참석자들이 휴대전화번호를 잘못된 형식으로 작성하여 저장되었습니다. 파이썬을 활용하여 방명록 명단을 확인하고, 올바른지 아닌지 휴대전화번호를 찾아내는 코드를 작성해주세요. [난이도 : ★★/5]

```
# 아래 목록을 방명록 명단으로 활용하세요!
김갑, 123456789
이을, 010-1234-5678
박병, 010-5678-111
최정, 111-1111-1111
정무, 010-3333-3333
```

- 함수에 방명록을 넣으면 txt 파일로 저장되게 해줍니다.
- 제대로 적은 방명록의 조건은 다음과 같습니다
 - 010 으로 시작함
 - 번호가 - 로 구분이 되어 있음
 - 를 포함하여 길이가 13임

✓ 보조 설명 :

1. 방명록 명단을 리스트 형태로 저장합니다. 각 요소는 참석자의 정보를 담고 있어야 합니다.
2. 방명록 명단을 순회하면서 각 참석자의 휴대전화번호를 확인합니다.
3. 휴대전화번호의 길이를 확인하여 올바른 형식인지 판단합니다.
4. 휴대전화번호가 '-' 문자로 구분된 3개의 그룹으로 나뉘어져 있는지 확인합니다.

5. 각 그룹의 길이가 맞는지 확인합니다. 첫번째 그룹은 "010"이고, 나머지 두 그룹은 4자리씩 구성되어야 합니다.

6. 모든 조건을 만족하지 않는 참석자의 정보를 출력합니다.

▼출력 예시

```
guest_book = ""김갑,123456789
이을,010-1234-5678
박병,010-5678-111
최정,111-1111-1111
정무,010-3333-3333""
wrong_guest_book(guest_book)
```

잘못 쓴 사람: 김갑

잘못 쓴 번호: 123456789

잘못 쓴 사람: 박병

잘못 쓴 번호: 010-5678-111

잘못 쓴 사람: 최정

잘못 쓴 번호: 111-1111-1111

📌 Q4. 한 IT 기업에서는 신규 직원의 신상정보를 확인하는 시스템을 도입하려고 합니다. 이 시스템은 주민등록번호를 받아 해당 직원이 몇년 몇월에 태어났는지 그리고 남자인지 여자인지를 출력하는 기능을 제공합니다. 파이썬을 활용해 다음의 조건을 충족하는 프로그램을 만들고, 효율적인 인사 관리 시스템을 만들어보세요. [난이도 : ★★★★★/5]

작성 조건

- 주민등록번호는 6자리 이후에 '-'로 구분되어야 하며, 총 길이는 '-'를 포함하여 14자리임
- 주민등록번호의 뒷자리 시작번호가 1 또는 3인 경우 해당 직원은 남성, 2 또는 4인 경우 해당 직원은 여성임
- 주민등록번호의 뒷자리 3, 4를 가질 수 있는 경우는 2000년생 이후 출생자 뿐임
- 주어진 주민등록번호가 조건을 만족하지 않을 경우, "잘못된 번호입니다"라는 메시지를 출력함

✓ 보조 설명 :

주민등록번호는 우리나라에서 사용하는 신분증 번호로, 생년월일과 성별 정보가 포함되어 있습니다. 이 문제에서는 주민번호를 입력하면 해당 개인이 언제 태어났는지, 그리고 남성인지 여성인지를 출력하는 함수를 만들어보려고 합니다.

이 문제를 해결하기 위해서는 입력된 주민등록번호를 분석하여 생년월일과 성별을 정확히 판별하는 함수를 작성해야 합니다. 조건을 하나씩 확인하면서 주민등록번호의 유효성을 검사

하고, 결과를 출력하는 로직을 구현해보세요.

구현 순서는 다음과 같습니다 :

1. 사용자로부터 주민등록번호를 입력받습니다.
2. 입력된 주민등록번호의 길이와 형식을 확인합니다.
 - 만약 길이와 형식이 맞지 않을 경우, "잘못된 번호입니다" 메시지를 출력하고 프로그램을 종료합니다.
3. 주민등록번호의 뒷자리 시작번호를 확인하여 성별을 판별합니다.
4. 주민등록번호의 앞 2자리가 00부터 23까지인 경우, 출생년도 확인이 필요합니다.
 - 이 경우 사용자에게 "2000년 이후 출생자입니까? (O/X)"와 같은 질문을 출력합니다.
 - 사용자의 입력을 받아 맞으면 "20###"년생, 틀리면 "19###"년생으로 저장합니다.
5. 최종적으로 생년월일과 성별을 출력합니다.

▼ 출력 예시

```
a = "500629-2222222"
check_id(a)
1950년06월 여자

a = "000629-2222222"
check_id(a)
2000년 이후 출생자 입니까? 맞으면 o 아니면 x : o
잘못된 번호입니다.
올바른 번호를 넣어주세요

a = "000629-2222222"
check_id(a)
2000년 이후 출생자 입니까? 맞으면 o 아니면 x : x
1900년06월 여자
```