

# 파이썬 프로그래밍

문제 1: 지정된 형식으로 숫자를 출력하는 함수, `print_num()` 설계

2020년 12월 10일 목

컴퓨터공학과

2019305059

이현수

## ■ 문제 1: 지정된 형식으로 숫자를 출력하는 함수, print\_num() 설계

- 1) 1,000 단위로 comma넣은 숫자를 표현 후 연이어 괄호 안에는 지정된 형식의 한글과 섞어 표현.
- 2) '만' 이상 단위는 숫자로 표현. 하지만 이때는 ','로 자릿수를 구분하지 않는다.
- 3) 하위 4자리, 0~9999는 한글과 숫자를 섞어서 표현한다.
- 4) 한글 단위 앞에 0이 오면 그 한글 단위는 출력되지 않는다.
- 5) 한글 단위 앞에 1이 오면 '1'자는 표시되지 않지만 그 한글 단위는 출력된다.

## ■ 코드

```
1 import os
2 import time
3 sz = os.path.getsize(__file__)
4 print(f'Program Size = {sz:#,}\n') #단계1. 파일크기 출력
5
6 def print_num(num):
7     str_num = str(num) #정수를 문자열로 변환
8     num_len = len(str_num) #문자열 길이
9     if num_len < 4: print(str_num, end='') #단계2. 1,000 단위 쉼표(,) 넣기 ...문자열길이가 4미만이라면 그냥출력
10    else: #문자열길이가 4 이상이라면
11        numlist=list(str_num) #문자열을 리스트로 변환
12        index = num_len - 3
13        numlist.insert(index, ",") #, 삽입
14        while True:
15            index = index - 3 #인덱스 3빼기
16            if index<=0: #인덱스가 맨앞자리거나 음수면 반복문 빠져감
17                break
18            else:
19                numlist.insert(index, ",") #아니면 인덱스에 , 삽입
20        str_num2=''.join(numlist) #리스트를 문자열로 변환
21        print(str_num2, end='') #출력
22    if str_num == '0': #단계3. 10000 이하의 자릿수에 대한 아라비아 숫자와 한글표현 ... str_num이 0이라면
23        print('()') #()만 출력
24    elif num_len==1: #숫자길이가 1이라면
25        print('{0}'.format(str_num)) #{숫자} 출력
26    elif num_len==2: #숫자길이가 2라면
27        print('(', end='')
28        if str_num[0]!='1': print(str_num[0], end='') #십의자리가 1이 아닐때 십의자리 숫자출력
29        print('십', end='') #'십' 출력
30        if str_num[1]!='0': print(str_num[1], end='') #일의자리가 0이 아닐때 일의자리
31        print(')')
32    elif num_len==3: #숫자길이가 3이라면
33        print('(', end='')
34        if str_num[0]!='1': print(str_num[0], end='') #백의자리가 '1'이 아니면 백의자리 숫자 출력
35        print('백', end='') #'백' 출력
36        if str_num[1]!='0': #십의자리숫자가 '0'이 아닐때
37            if str_num[1]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면 '1'일때 '십'만 출력
38            else: print(str_num[1]+'십', end='') #그외 숫자+'십' 형태로 출력
39        if str_num[2]!='0': print(str_num[2], end='') #일의자리숫자가 '0'이 아니면 일의자리숫자 그대로 출력
40        print(')')
```

파일크기 출력  
(주석문포함 크기)

1000단위로 comma  
넣는 코드

지정형식으로 한글섞어  
표현

```

41 elif num_len==4: #숫자길이가 4라면
42     print('(', end='')
43     if str_num[0] != '1': print(str_num[0], end='') #천의자리숫자가 '1'이 아니면 천의자리 숫자 출력
44     print('천', end='') #천 출력
45     if str_num[1] != '0': #백의자리가 '0'이 아닐때
46         if str_num[1]=='1': print('백', end='') #백의자리숫자가 '0'이 아니면서 '1'일때 '백'만 출력
47         else: print(str_num[1] + '백', end='') #그외 숫자+ '백' 형태로 출력
48     if str_num[2] != '0': #십의자리숫자가 '0'이 아닐때
49         if str_num[2]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면서 '1'일때 '십'만 출력
50         else: print(str_num[2] + '십', end='') #십의자리숫자가 '0'이 아니면 숫자+ '십' 형태로 출력
51     if str_num[3] != '0': print(str_num[3], end='') #일의자리숫자가 '0'이 아니면 일의자리숫자 그대로 출력
52     print(')')
53 else: #숫자길이가 5이상이라면
54     print('(', end='')
55     if(num_len==5): #숫자길이가 5일경우
56         if str_num[0]!='1': print(str_num[0], end='') #만의자리수자가 '1'이 아니면
57         print('만', end='') #만 출력
58     else: print(str_num[:len(str_num)-4] + '만', end='') #길이6 이상일때 인덱스 부터 만의자리숫자까지 출력후 '만' 출력
59     if str_num[len(str_num)-4] != '0': #천의자리가 '0'이 아닐때
60         if str_num[len(str_num)-4]=='1': print('천', end='') #천의자리숫자가 '1'이 아니면서 '1'일때 '천'만 출력
61         else: print(str_num[len(str_num)-4] + '천', end='') #천의자리숫자가 '0'이 아니면 숫자+ '천' 형태로 출력
62     if str_num[len(str_num)-3] != '0': #백의자리가 '0'이 아닐때
63         if str_num[len(str_num)-3]=='1': print('백', end='') #백의자리숫자가 '0'이 아니면서 '1'일때 '백'만 출력
64         else: print(str_num[len(str_num)-3] + '백', end='') #백의자리숫자가 '0'이 아니면 숫자+ '백' 형태로 출력
65     if str_num[len(str_num)-2] != '0': #십의자리숫자가 '0'이 아닐때
66         if str_num[len(str_num)-2]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면서 '1'일때 '십'만 출력
67         else: print(str_num[len(str_num)-2] + '십', end='') #십의자리숫자가 '0'이 아니면 숫자+ '십' 형태로 출력
68     if str_num[len(str_num)-1] != '0': print(str_num[len(str_num)-1], end='') #일의자리숫자가 '0'이 아니면 숫자 그대로 출력
69     print(')')

```

지정된형식으로 한글섞어  
표현

```

71 int_list = [12334567891, 20500, 1100007, 900001, 9028, 100, 15, 8, 0]

```

```

73 s_time = time.time() #단계4. 시간측정시작

```

```

75 for i in int_list:

```

```

76     print_num(i) #출력

```

```

77 # num_lines = 64

```

```

78 c_time = time.time() #시간측정 중단

```

```

79 print('\nTotal Execution Time = {0:.4f}[sec]'.format(c_time - s_time)) #시간측정 결과 출력

```

시간측정 및 테스트

Print\_num함수 num\_lines = 64

## ■ 코드설명

```
1 import os
2 import time
3 sz = os.path.getsize(__file__)
4 print(f'Program Size = {sz:#{}}\n') #단계1. 파일크기 출력
```

os모듈을 사용해서 파일 사이즈를 구한다.

```
6 def print_num(num):
7     str_num = str(num) #정수를 문자열로 변환
8     num_len = len(str_num) #문자열 길이
9     if num_len < 4: print(str_num, end='') #단계2. 1,000 단위 쉼표(,) 넣기 ... 문자열길이가 4미만이라면 그냥출력
10    else: #문자열길이가 4이상이라면
11        numlist=list(str_num) #문자열을 리스트로 변환
12        index = num_len - 3
13        numlist.insert(index, ",") # , 삽입
14        while True:
15            index = index - 3 #인덱스 3빼기
16            if index<=0: #인덱스가 맨앞자리거나 음수면 반복문 빠져나감
17                break
18            else:
19                numlist.insert(index, ",") #아니면 인덱스에 , 삽입
20        str_num2=''.join(numlist) #리스트를 문자열로 변환
21        print(str_num2, end='') #출력
```

num 매개변수를 가지는 print\_num함수를 작성한다. 사용자로부터 매개변수 받은 num 정수 변수를 str()을 통해 문자열로 변경해 str\_num에 저장한다. 그리고 len()을 이용해 str\_num의 문자열 길이를 구해 num\_len에 저장한다. (코드9~) if조건문을 사용해 문자열길이가 4미만인 것과 아닌것으로 구분해, 4미만인 경우 그대로 출력해준다. (코드10~) 문자열길이가 4이상이라면 ,를 찍어줘야 한다. list()를 이용해 str\_num을 리스트형태로 만들어 numlist에 저장한다. 그리고 index = num\_len-3을 통해 ,를 추가할 인덱스를 구한다. 그 후 numlist에 insert메소드를 이용해 index 위치에 ' , '를 삽입한다. 그 후 while True:를 통해 무한반복문을 만들어 index가 0이하일 때 빠져나온다. 다음 자리에 ' , '를 삽입하기 위해서 index = index -3을 해주고 numlist리스트에 insert를 통해 index인덱스에 ' , '를 삽입한다. 그리고 join함수를 통해 numlist리스트를 문자열로 만들어 str\_num2로 저장 후 출력한다.

```
22 if str_num == '0': #단계3. 10000 이하의 자릿수에 대한 아라비아 숫자와 한글표현 ... str_num이 0이라면
23     print('()') #()만 출력
24 elif num_len==1: #숫자길이가 1이라면
25     print('{0}'.format(str_num)) #(숫자) 출력
26 elif num_len==2: #숫자길이가 2라면
27     print(',', end='')
28     if str_num[0]!='1': print(str_num[0], end='') #십의자리가 1이 아닐때 십의자리 숫자출력
29     print('십', end='') #'십' 출력
30     if str_num[1]!='0': print(str_num[1], end='') #일의자리가 0이 아닐때 일의자리 숫자출력
31     print('')
```

괄호안에 한글과 숫자를 섞어 적는다. str\_num이 '0'일 경우에 일의자리와 상관없이 ()를 출력하기 때문에 처음 if 조건문에 조건을 걸어둔다. 그리고 num\_len이 1, 2, 3, 4, 그 이상인 경우로 나눈다.

num\_len이 1인 경우, 괄호안에 그냥 숫자를 출력한다.

num\_len이 2인 경우, str\_num[0](십의자리)가 '1'이 아니면 str\_num[0]을 출력하고, '십'은 무조건 출력한다. str\_num[1]이 '0'이 아니라면 str\_num[1]을 무조건 출력시킨다.

```

32 elif num_len==3: #숫자길이가 3이라면
33     print('(', end='')
34     if str_num[0]!='1': print(str_num[0], end='') #백의자리가 '1'이 아니면 백의자리 숫자 출력
35     print('백', end='') # '백' 출력
36     if str_num[1]!='0': #십의자리숫자가 '0'이 아닐때
37         if str_num[1]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면서 '1'일때 '십'만 출력
38         else: print(str_num[1]+'십', end='') #그외 숫자+ '십' 형태로 출력
39     if str_num[2]!='0': print(str_num[2], end='') #일의자리숫자가 '0'이 아니면 일의자리숫자 그대로 출력
40     print(')')

```

num\_len이 3인 경우, str\_num[0](백의자리)가 '1'이 아니면 str\_num[0]을 출력하고 '백'은 무조건 출력한다. str\_num[1](십의자리)가 '0'이 아닐 때 str\_num[1]이 '1'이라면 '십'만 출력하고 아니면 str\_num[1]과 '십'을 출력한다. 그리고 str\_num[2]가 '0'이 아니면 str\_num[2]을 출력한다.

```

41 elif num_len==4: #숫자길이가 4라면
42     print('(', end='')
43     if str_num[0] != '1': print(str_num[0], end='') #천의자리숫자가 '1'이 아니면 천의자리 숫자 출력
44     print('천', end='') # '천' 출력
45     if str_num[1] != '0': #백의자리가 '0'이 아닐때
46         if str_num[1]=='1': print('백', end='') #백의자리숫자가 '0'이 아니면서 '1'일때 '백'만 출력
47         else: print(str_num[1] + '백', end='') #그외 숫자+ '백' 형태로 출력
48     if str_num[2] != '0': #십의자리숫자가 '0'이 아닐때
49         if str_num[2]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면서 '1'일때 '십'만 출력
50         else: print(str_num[2] + '십', end='') #십의자리숫자가 '0'이 아니면 숫자+ '십' 형태로 출력
51     if str_num[3] != '0': print(str_num[3], end='') #일의자리숫자가 '0'이 아니면 일의자리숫자 그대로 출력
52     print(')')

```

num\_len이 4인 경우, str\_num[0](천의자리)가 '1'이 아니면 str\_num[0]을 출력하고 '천'은 무조건 출력한다. str\_num[1](백의자리)가 '0'이 아닐 때 만약 str\_num[1]이 '1'이면 '백'만 출력하고 아니면 str\_num[1]과 '백'을 출력한다.

str\_num[2](십의자리)가 '0'이 아닐 때 만약 str\_num[2]가 '1'이면 '십'만 출력하고 아니면 str\_num[2]과 '십'을 출력한다.

str\_num[3](일의자리)가 '0'이 아닐 때 str\_num[3]을 출력한다.

```

53 else: #숫자길이가 5이상이라면
54     print('(', end='')
55     if(num_len==5): #숫자길이가 5일경우
56         if str_num[0]!='1': print(str_num[0], end='') #만의자리수사가 '1'이 아니면 만의자리숫자출력
57         print('만', end='') # '만' 출력
58     else: print(str_num[:len(str_num)-4] + '만', end='') #길이6이상일때 인덱스0부터 만의자리숫자까지 출력후 '만' 출력
59     if str_num[len(str_num)-4] != '0': #천의자리가 '0'이 아닐때
60         if str_num[len(str_num)-4]=='1': print('천', end='') #천의자리숫자가 '0'이 아니면서 '1'일때 '천'만 출력
61         else: print(str_num[len(str_num)-4] + '천', end='') #천의자리숫자가 '0'이 아니면 숫자+ '천' 형태로 출력
62     if str_num[len(str_num)-3] != '0': #백의자리가 '0'이 아닐때
63         if str_num[len(str_num)-3]=='1': print('백', end='') #백의자리숫자가 '0'이 아니면서 '1'일때 '백'만 출력
64         else: print(str_num[len(str_num)-3] + '백', end='') #백의자리숫자가 '0'이 아니면 숫자+ '백' 형태로 출력
65     if str_num[len(str_num)-2] != '0': #십의자리숫자가 '0'이 아닐때
66         if str_num[len(str_num)-2]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면서 '1'일때 '십'만 출력
67         else: print(str_num[len(str_num)-2] + '십', end='') #십의자리숫자가 '0'이 아니면 숫자+ '십' 형태로 출력
68     if str_num[len(str_num)-1] != '0': print(str_num[len(str_num)-1], end='') #일의자리숫자가 '0'이 아니면 숫자 그대로 출력
69     print(')')

```

그외 num\_len이 5이상인 경우, 만이상을 출력하는 코드는 num\_len이 5인 경우와 6이상으로 다시 나눈다. 그리고 천단위 아래로 출력하는 코드는 공통으로 공유한다.

5인 경우, str\_num[0]이 '1'이 아니면 str\_num[0]을 출력 후 '만'은 무조건 출력한다.

6이상인 경우, str\_num[:len(str\_num)-4]+'만'을 출력한다.

str\_num[:len(str\_num)-4]의 의미는 str\_num의 인덱스 0부터 len(str\_num)-4-1까지 문자열이다.

```

53     else: #숫자길이가 5 이상이라면
54         print('(', end='')
55         if(num_len==5): #숫자길이가 5일경우
56             if str_num[0]!='1': print(str_num[0], end='') #만의자리숫자가 '1' 이 아니면 만의자리숫자출력
57             print('만',end='') #'만' 출력
58         else: print(str_num[:len(str_num)-4] + '만', end='') #길이6 이상일때 인덱스0 부터 만의자리숫자까지 출력후 '만' 출력
59         if str_num[len(str_num)-4] != '0':#천의자리가 '0' 이 아닐때
60             if str_num[len(str_num)-4]=='1': print('천', end='')#천의자리숫자가 '0' 이 아니면서 '1' 일때 '천' 만 출력
61             else: print(str_num[len(str_num)-4] + '천', end='')#천의자리숫자가 '0' 이 아니면 숫자+ '천' 형태로 출력
62         if str_num[len(str_num)-3] != '0':#백의자리가 '0' 이 아닐때
63             if str_num[len(str_num)-3]=='1': print('백', end='')#백의자리숫자가 '0' 이 아니면서 '1' 일때 '백' 만 출력
64             else: print(str_num[len(str_num)-3] + '백', end='')#백의자리숫자가 '0' 이 아니면 숫자+ '백' 형태로 출력
65         if str_num[len(str_num)-2] != '0':#십의자리숫자가 '0' 이 아닐때
66             if str_num[len(str_num)-2]=='1': print('십', end='')#십의자리숫자가 '0' 이 아니면서 '1' 일때 '십' 만 출력
67             else: print(str_num[len(str_num)-2] + '십', end='')#십의자리숫자가 '0' 이 아니면 숫자+ '십' 형태로 출력
68         if str_num[len(str_num)-1] != '0': print(str_num[len(str_num)-1], end='')#일의자리숫자가 '0' 이 아니면 숫자 그대로 출력
69         print(')')

```

그 후 str\_num[:len(str\_num)-4](천의자리)가 '0'이 아닐 때 str\_num[:len(str\_num)-4]이 '1'이라면 '천'만 출력하고, 그 외의 경우 str\_num[:len(str\_num)-4]와 '천'을 출력한다.

str\_num[:len(str\_num)-3](백의자리)가 '0'이 아닐 때 str\_num[:len(str\_num)-3]이 '1'이라면 '백'만 출력하고, 그외의 경우 str\_num[:len(str\_num)-3]와 '백'을 출력한다.

str\_num[:len(str\_num)-2](십의자리)가 '0'이 아닐 때 str\_num[:len(str\_num)-2]이 '1'이라면 '십'만 출력하고, 그외의 경우 str\_num[:len(str\_num)-2]와 '십'을 출력한다.

str\_num[:len(str\_num)-1](일의자리)가 '0'이 아니면 str\_num[:len(str\_num)-1]을 출력한다.

```

71 int_list = [12334567891, 20500, 1100007, 900001, 9028, 100, 15, 8, 0]
72
73 s_time = time.time() #단계4. 시간측정시작
74
75 for i in int_list:
76     print_num(i) #출력
77 # num_lines = 64
78 c_time = time.time() #시간측정 중단
79 print('\nTotal Execution Time = {0:.4f}[sec]'.format(c_time - s_time)) #시간측정 결과 출력

```

int\_list 배열에 출력할 정수숫자를 입력한다.

s\_time = time.time()을 통해 시간측정을 시작한다.

for반복문을 통해 int\_list에 있는 정수 한 개씩 print\_num함수를 통해 출력시킨다.

그후 c\_time = time.time()을 통해 시간측정을 중단한다. 그리고 c\_time - s\_time을 통해 시간측정 결과를 출력한다.

## ■ 실행

```
int_list = [12334567891, 20500, 1100007, 900001, 9028, 100, 15, 8, 0]
```

```
이현수1 ×
"C:\Program Files\Python38\python.exe" C:/Users/nec/Desktop/파이썬프로그래밍과제2/prob1/이현수1.py
Program Size = 5,233

12,334,567,891(1233456만7천8백9십1)
20,500(2만5백)
1,100,007(110만7)
900,001(90만1)
9,028(9천2십8)
100(백)
15(십5)
8(8)
0()

Total Execution Time = 0.0000[sec]

Process finished with exit code 0
```

레포트 공고 예제 출력과 똑같이 출력된다.

```
int_list = [1000100, 11520, 4502107, 901001, 9128, 310, 25, 7, 1]
```

```
이현수1 ×
"C:\Program Files\Python38\python.exe" C:/Users/nec/Desktop/파이썬프로그래밍과제2/prob1/이현수1.py
Program Size = 5,229

1,000,100(100만백)
11,520(만천5백2십)
4,502,107(450만2천백7)
901,001(90만천1)
9,128(9천백2십8)
310(3백십)
25(2십5)
7(7)
1(1)

Total Execution Time = 0.0000[sec]

Process finished with exit code 0
```

다른 값으로 실행한 결과 정상적으로 출력된다.

## ■교훈 및 시행착오

1차과제에서 숫자에 ' , '찍기 내용이 있어서 1차과제에서 했던 소스를 이용해 비교적 수월하게 시작했다.

코드를 작성하고 실험하는 과정에서 2100의 경우 2,100(2천1백)처럼 1백이 출력되는 오류를 발견했다.

그래서 조건문을 더 추가해서 올바르게 수정했다.