

▼ 002

'인생은 짧으니 파이썬이 [...]'

[illegible]

▼ 003

```

1 #re_nomal.sample.py
2 data = """
3 홍길동의 주민번호는 800905-1049118 입니다.
4 그리고 고길동의 주민번호는 700905-1059119 입니다.
5 그렇다면 누가 형님일까요?
6 """
7 print(data)

```

홍길동의 주민번호는 800905-1049118 입니다.
 그리고 고길동의 주민번호는 700905-1059119 입니다.
 그렇다면 누가 형님일까요?

```

1 result = []
2 for line in data.split("\n"):
3     word_result = []
4     for word in line.split(" "):
5         if len(word) == 14 and word[:6].isdigit() and word[7:].isdigit():
6             word = word[:6] + "-" + "*****"
7             word_result.append(word)
8     result.append(" ".join(word_result))
9 print("\n".join(result))

```

홍길동의 주민번호는 800905-***** 입니다.
 그리고 고길동의 주민번호는 700905-***** 입니다.
 그렇다면 누가 형님일까요?

```

1 import re
2
3 pattern = re.compile("(Wd{6})[-]Wd{7}")
4 print(pattern.sub("Wg<1>-*****", data))

```

홍길동의 주민번호는 800905-***** 입니다.
 그리고 고길동의 주민번호는 700905-***** 입니다.
 그렇다면 누가 형님일까요?

```

1 pattern = re.compile("(Wd{6})[-]Wd{7}")
2 print(pattern.sub("Wg<1>-abcdefg", data))

```

홍길동의 주민번호는 800905-abcdefg 입니다.
 그리고 고길동의 주민번호는 700905-abcdefg 입니다.
 그렇다면 누가 형님일까요?

▼ 004

```

1 %%writefile struct_sample.c
2
3 #include <stdio.h>
4 typedef struct {
5     double v;
6     int t;
7     char c;
8 } save_type;
9
10 int main() {
11     save_type s = {7.5f, 15, 'A'};
12     FILE *fp = fopen("output", "w");
13     fwrite(&s, sizeof(save_type), 1, fp);
14     printf("%d\n", struct_sample)
15     fclose(fp);
16     return 0;
17 }

```

Overwriting struct_sample.c

```

1 # 느낌표를 입력하면 리눅스 명령어 쓸 수 있게 해줌

```

```
1 !ls -al
```

```
total 20
drwxr-xr-x 1 root root 4096 Mar 14 01:22 .
drwxr-xr-x 1 root root 4096 Mar 14 00:13 ..
drwxr-xr-x 4 root root 4096 Mar 10 20:50 .config
drwxr-xr-x 1 root root 4096 Mar 10 20:51 sample_data
-rw-r--r-- 1 root root 247 Mar 14 01:22 struct_sample.c
```

```
1 !cat struct_sample.c
```

```
#include <stdio.h>
typedef struct {
    double v;
    int t;
    char c;
} save_type;

int main() {
    save_type s = {7.5f, 15, 'A'};
    FILE *fp = fopen("output", "w");
    fwrite(&s, sizeof(save_type), 1, fp);
    fclose(fp);
    return 0;
}
```

```
1 !pwd
```

/content

```
1 !gcc struct_sample.c -o struct_sample
```

```
struct_sample.c: In function 'main' :
struct_sample.c:13:19: error: 'struct_sample' undeclared (first use in this function)
13 |     printf("%d\n",struct_sample)
    |                      ^~~~~~
struct_sample.c:13:19: note: each undeclared identifier is reported only once for each function it appears in
struct_sample.c:13:33: error: expected ';' before 'fclose'
13 |     printf("%d\n",struct_sample)
    |                               ^
14 |     fclose(fp);
    |     ~~~~~
```

```
1 !struct_sample
```

/bin/bash: struct_sample: command not found

```
1 !./struct_sample
```

```
1 import struct
2
3 with open('output','rb') as f:
4     chunk=f.read(16)
5     result=struct.unpack('dcccc',chunk)
6 print(result)

(7.5, 15, b'A', b'U', b'Wx00', b'Wx00')
```

```
1
```

005

```
1 import datetime
2
3 day1=datetime.date(2019,12,14)
4 day2=datetime.date(2021,6,5)

1 type(day1)

datetime.date

1 diff=day2-day1
2 diff.days
```

```
1 day3=datetime.datetime(2020, 12, 14, 14, 10, 50)
```

```
1 day3.day
```

```
14
```

```
1 day3.year
```

```
2020
```

```
1 day3.minute
```

```
10
```

```
1 date=datetime.date(2019, 12, 14)
2 time=datetime.time(10,43,50)
3
4 dt=datetime.datetime.combine(date,time)
```

```
1 dt
```

```
datetime.datetime(2019, 12, 14, 10, 43, 50)
```

```
1 datetime.date(2023,3,14).weekday()
2 # 기준: monday == 0 / 단, weekday일때만! isoweekday는 월요일이 1임. this is standard.
```

```
1
```

```
1 datetime.date(2023,3,14).isoweekday()
```

```
2
```

```
1 wd=['mon','tue','wed','thu','fri','sat','sun']
```

```
1 wd[datetime.date(2023,3,14).weekday()]
```

```
'tue'
```

```
1 datetime.date.today().weekday()
```

```
1
```

▼ 006

[+ 코드](#)
[+ 텍스트](#)

```
1 import datetime
2
3 datetime.date.today() + datetime.timedelta(days=2,hours=100)
```

```
datetime.date(2023, 3, 20)
```

```
1 import calendar
2
3 calendar.isleap(100)
```

```
False
```

```
1 from collections import deque
2 a=[1,2,3,4,5]
3 q=deque(a)
4 q.rotate(2)
5 result=list(q) #deque형태를 함수를 이용해서 다시 list 로 변환.
6 result
```

```
[4, 5, 1, 2, 3]
```

```
1 l=[1,2,3,4,5]
2 l.append(6)
```

```
1 l
```

```
[1, 2, 3, 4, 5, 6]
```

```
1 l.pop(0)
```

```
1
```

```
1 l
```

```
[2, 3, 4, 5, 6]
```

```
1 l.pop(3)
```

```
5
```

```
1 l
```

```
[2, 3, 4, 6]
```

```
1 l.pop() #그냥 pop쓰게 되면 맨 마지막 원소 제거하게 됨. 그리고 이름이 pop인 이유는 스택에서 push&pop named use.
```

```
4
```

