

/* elice */

도레미 파이썬

조건문 : 진실! 혹은 거짓?



이호준 선생님

오늘의 목표

컴퓨터에게 정보를 전달할 수 있다.

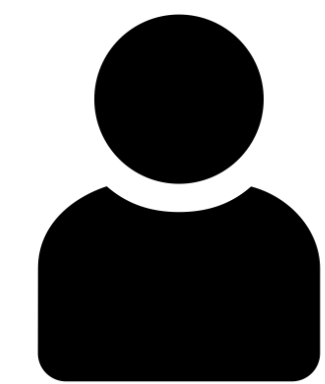
논리 자료형을 알고, 이를 활용할 수 있다.

조건에 따라 문제를 해결할 수 있다.

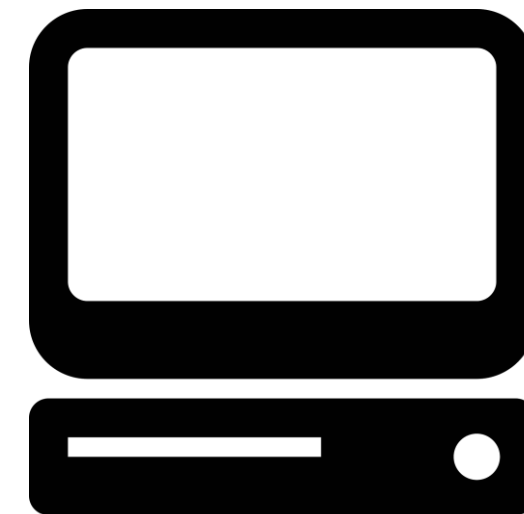
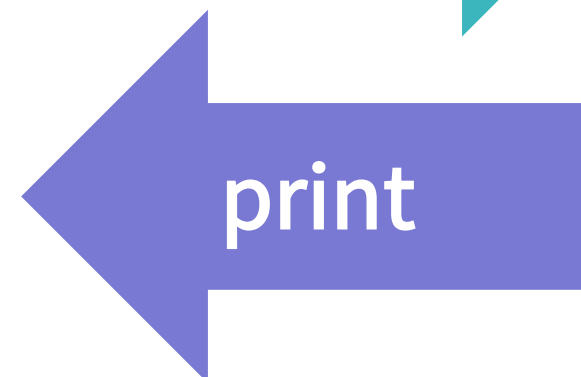
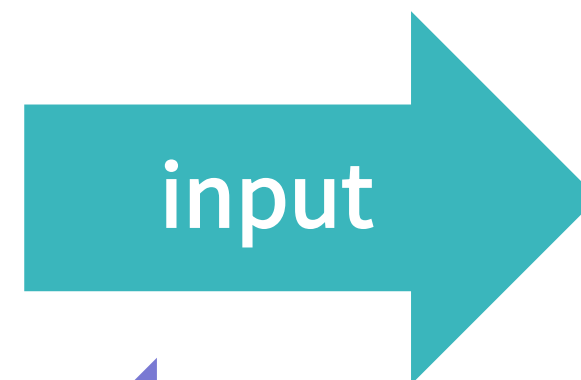
입력 – input()

입력

Print를 이용해 컴퓨터로부터
정보를 전달받았다면,
이제는 **정보를 컴퓨터에게 전달**해주자!



사용자

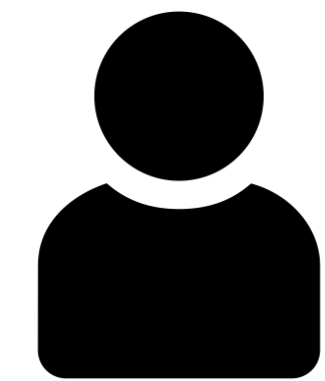


컴퓨터

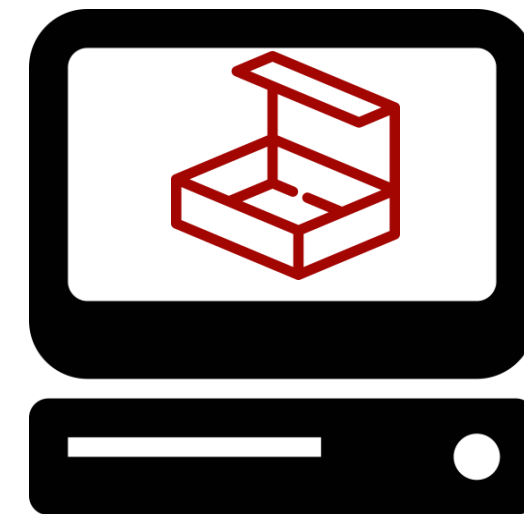
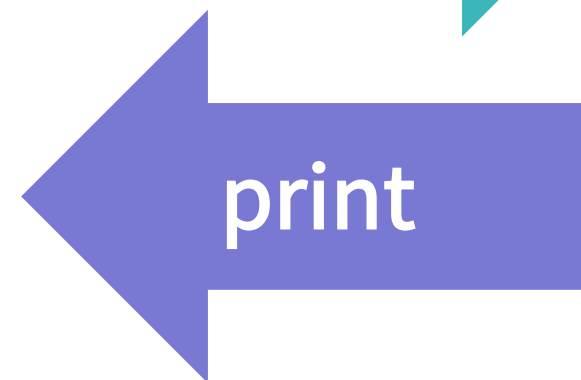
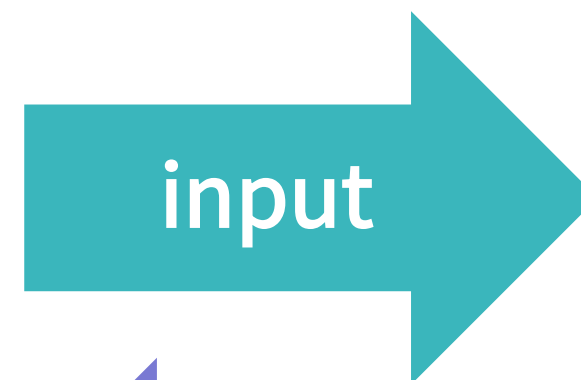
입력

이때, 컴퓨터는 사용자가 전달한 값을
어딘가에 보관해야 한다

→ 변수를 사용!



사용자



컴퓨터

input()

변수 = input()

```
var = input()
```

input()

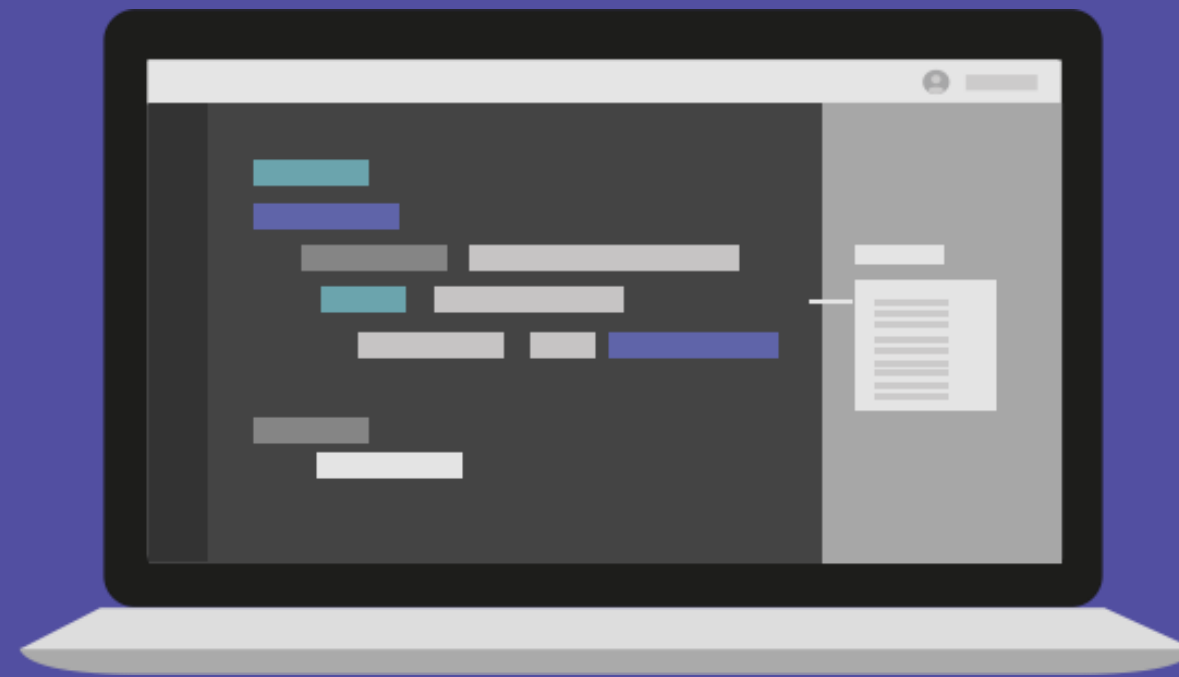
터미널에 값을 “입력” 해줍니다

```
/* 코드가 실행되는 중입니다! */
```

> 터미널 입력값을 넣어주세요.

[실습1]

따라쟁이 앵무새



input()의 중요한 특징

무엇을 입력하든 “문자열”로 입력 받아진다!

만약 숫자를 입력해야 한다면 어떻게

해야할까?

입력을 받은 후에 가공해주어야 한다!

어떻게 가공해주지?

문자열을 숫자로 바꾸고 싶어!

자료형 사이의 변환 → **형 변환!**

형 변환

바꿀_자료형(바뀔_자료)

integer #숫자(정수)

float #숫자(실수)

string #문자열

list #리스트

형 변환

```
a = '345'
```

```
b = int('345')
```

```
print(a,b) #345 345
```

```
print(type(a)) #<class 'str'> | 문자열
```

```
print(type(b)) #<class 'int'> | 숫자
```

[실습2]

2배로 적립



논리형 자료와 비교연산

1) 논리 자료형

참(True) 혹은 거짓(False)을 나타내는
자료형을 논리 자료형이라고 합니다

2) 비교 연산자

숫자나 문자의 값을 비교하는 연산자
주어진 진술이 참이면 True, 거짓이면 False

```
print(3 < 5)      #True  
print(7 == 5)     #False  
print(2 >= 10)    #False  
print(5 != 10)    #True
```


비교 연산자의 종류

== 같다

!= 다르다

> 왼쪽이 더 크다

< 오른쪽이 더 크다

>= 왼쪽이 같거나 크다

<= 오른쪽이 같거나 크다

[실습3]

명제 만들기



3) 논리 자료형의 연산

True, False밖에 없는 논리 자료형

→ 새로운 연산이 필요해!

1. AND(모두 True니?)

각 논리가 모두 True여야 True!

```
print(3==3 and 4<=5 and 6>2)
```

#세 항이 모두 True이므로, True!

```
>>> True
```

2. OR(True가 있니?)

논리들 중 True가 존재하면 True!

```
print(3==4 or 4<=5 or 6<2)
```

```
# 4<=5가 True이므로, True가 존재하기에 True!
```

```
>>> True
```

3. NOT(기존의 논리와 반대로!)

논리값을 뒤집는다!

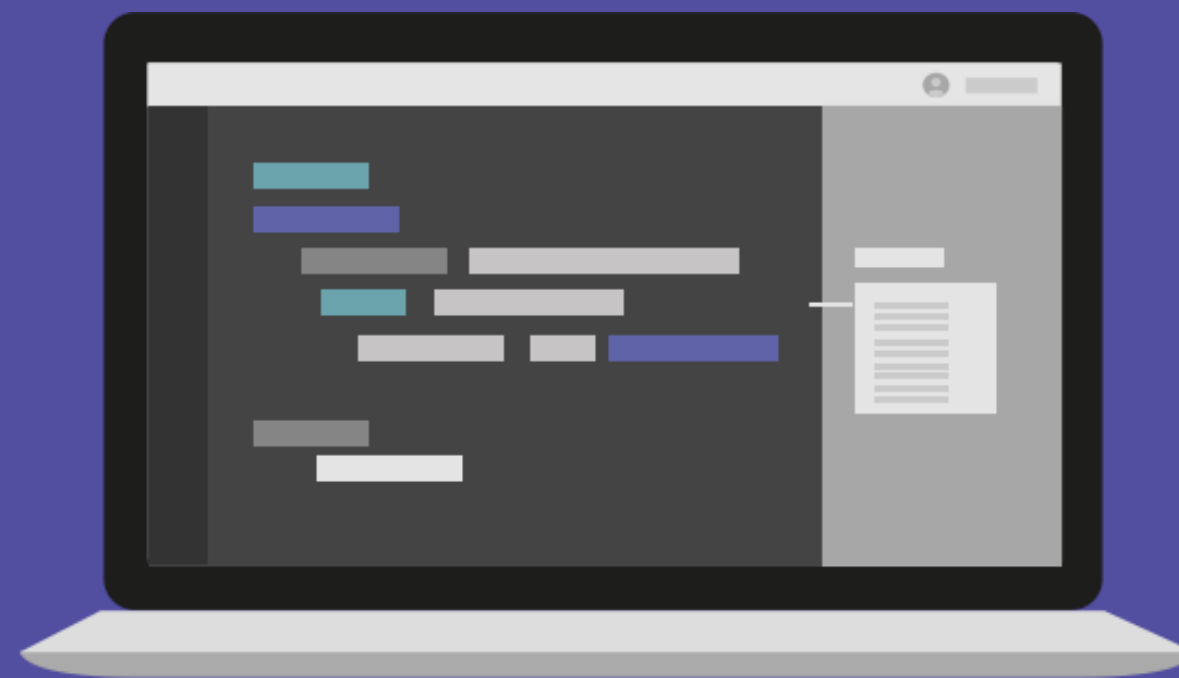
```
print(not 3==4)
```

```
# False에 Not을 붙였으므로, True!
```

```
>>> True
```

[실습4]

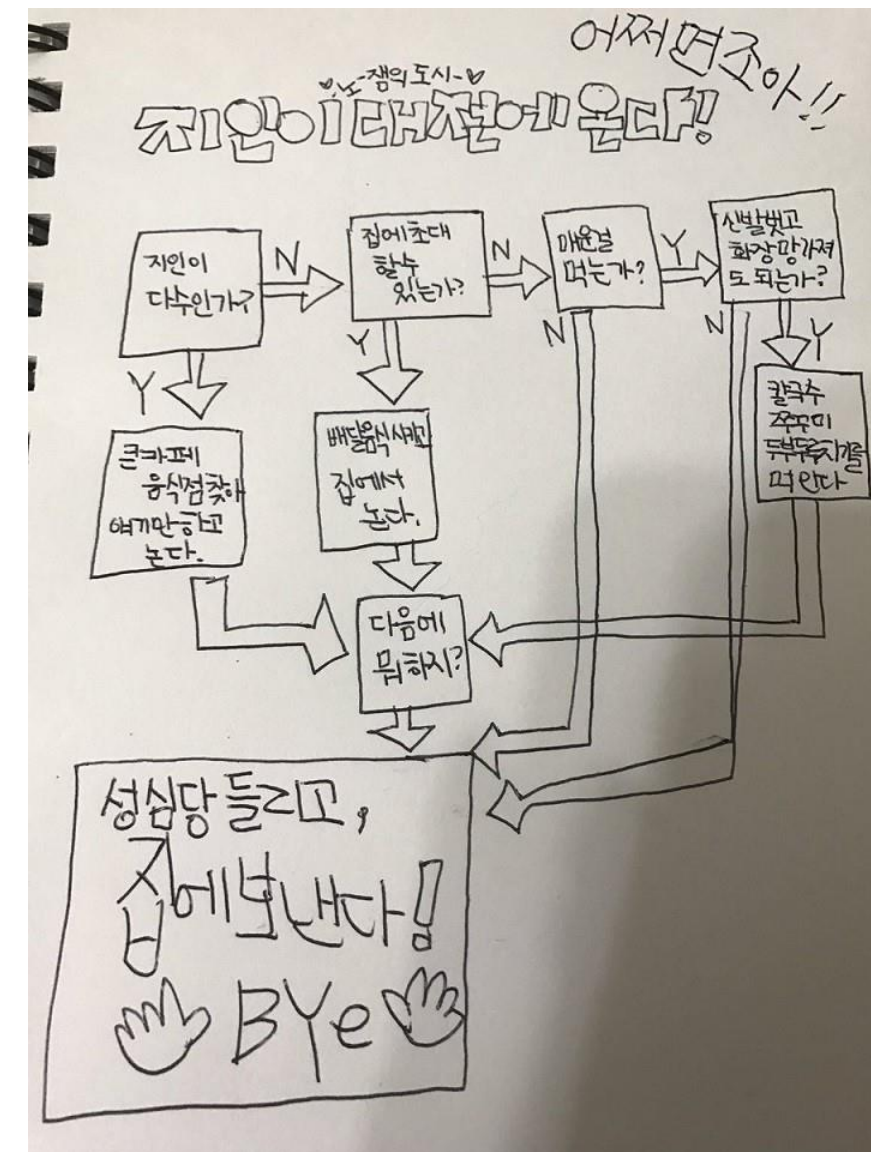
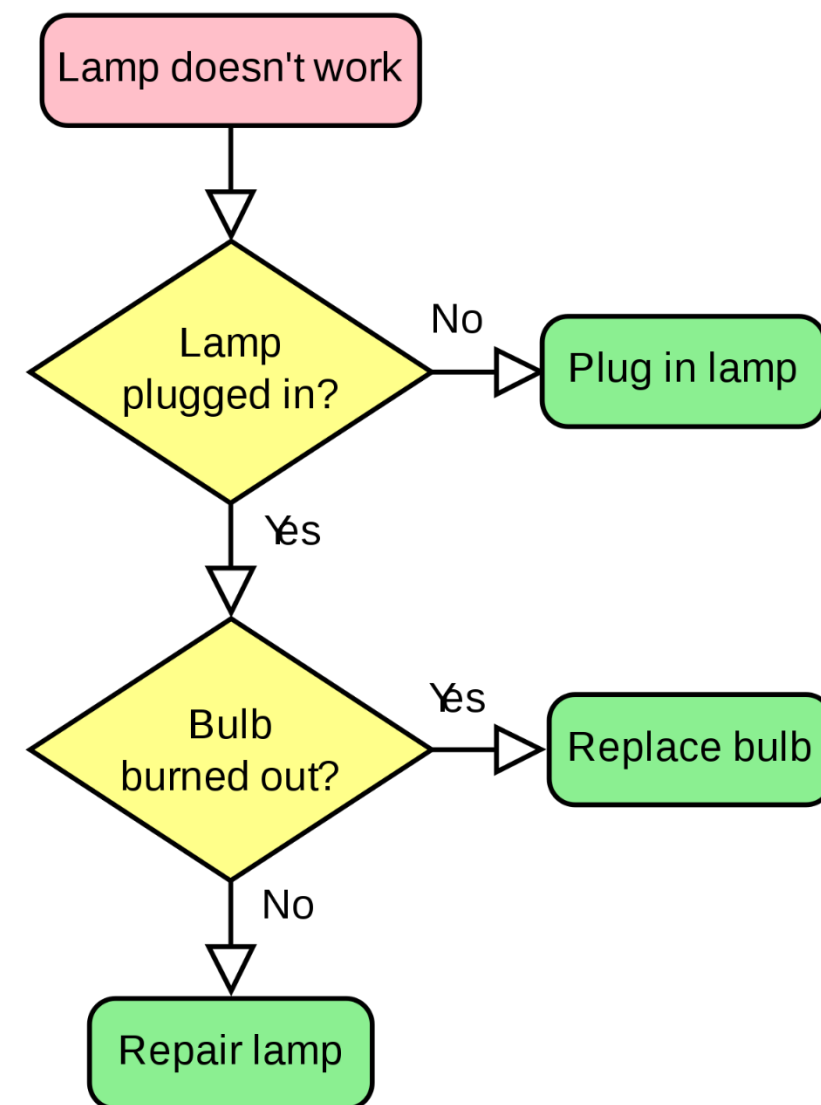
빈칸 추론



만약에 내가 간다면 - 조건문

왜 조건문이 필요할까?

우리 삶에는 **조건에 따라서** 해야 할 일이
다른 경우가 많기 때문이다!



조건문

어떠한 특정 조건에 따라서
실행되는 명령이 달라지는 구문!

파이썬의 조건문 - if 문

만약 $i == 1$ 이면, i 를 출력하라!

if

조건

명령

if 문

조건이 **True**일 때, **명령** 실행

```
if 조건:  
    <수행할 명령>  
    <수행할 명령>  
    .....
```

```
if string[0] == "a":  
    count = count + 1  
    print(string)
```

if 문

if문에 들어갈 명령들은 **같은 들여쓰기**로 구분!

```
if 조건:  
    __<수행할 명령>  
    __<수행할 명령>  
    __.....
```

```
if string[0] == "a":  
    __count = count + 1  
    __print(string)
```

if 문에서 조건을 만족하지 못하면?

만약 $i == 10$ 이면, i 를 출력하라!

아니면 $i + 1$ 을 출력해라!

else ($i \neq 1$)

if-else 문

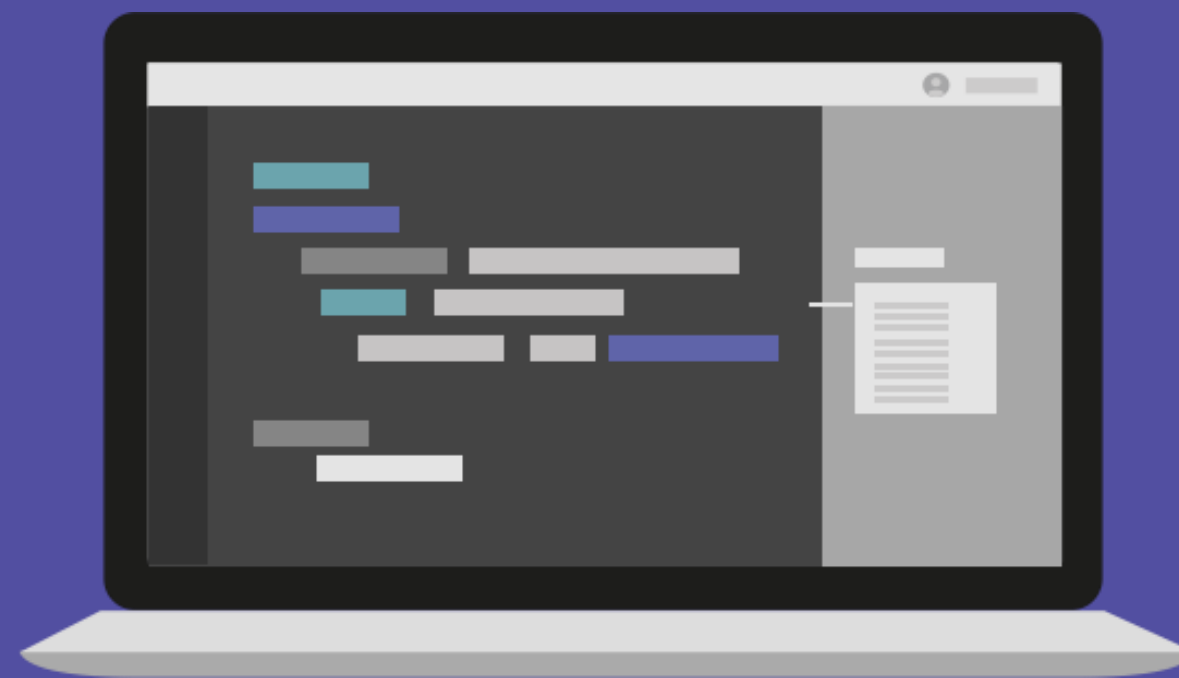
조건이 **True**면 if문 **False**면 else문 실행

```
if 조건:  
    <수행할 명령>  
else:  
    <수행할 명령>
```

```
x = input()  
if x in ['a', 'e', 'i', 'o',  
        'u']:  
    print("모음입니다.")  
else:  
    print("자음입니다.")
```

[실습5]

홀썩판별기



if 문에서 조건을 만족하지 못했을 때...

만약 점수가 90 이상이라면 A를 출력해라

아닌 경우에 75 이상이라면 B를 출력해라!

else if → elif

if-elif 문

조건 1이 **True**면 if문

조건 1이 **False**이면서 조건 2가 **True**면 elif문 실행

```
if 조건 1:  
    <수행할 명령>  
elif 조건 2:  
    <수행할 명령>
```

```
x = int(input())  
if x % 2 == 0:  
    print("2의 배수입니다.")  
elif x % 3 == 0:  
    print("3의 배수입니다.")
```

정리 : if-elif-else 문

```
if 조건 1:
```

```
    do A
```

```
elif 조건 2:
```

```
    do B
```

```
elif 조건 3:
```

```
    do C
```

```
    . . .
```

```
else:
```

```
    do X
```

조건 1 **True**

→ A 실행

조건1 **False** and 조건2 **True**

→ B 실행

조건1 **False** and 조건2 **False** and 조건3 **True**

→ C 실행

...

모든 조건이 **False**

→ X 실행

[실습6]

업-다운 게임



/* elice */

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice