



단국대학교
SW중심대학

창의적 사고와 코딩

제 5장 조건문

박 소 현

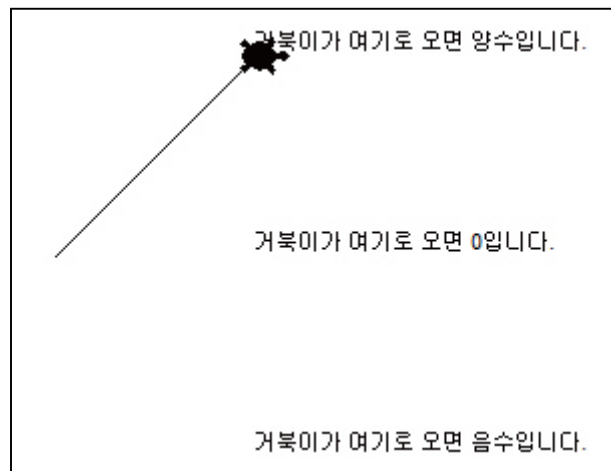
sohyunpark@dankook.ac.kr



(1) 터틀 그래픽을 사용하여 동전의 앞면이나 뒷면이 나오는 동전 던지기 게임을 작성해보자.

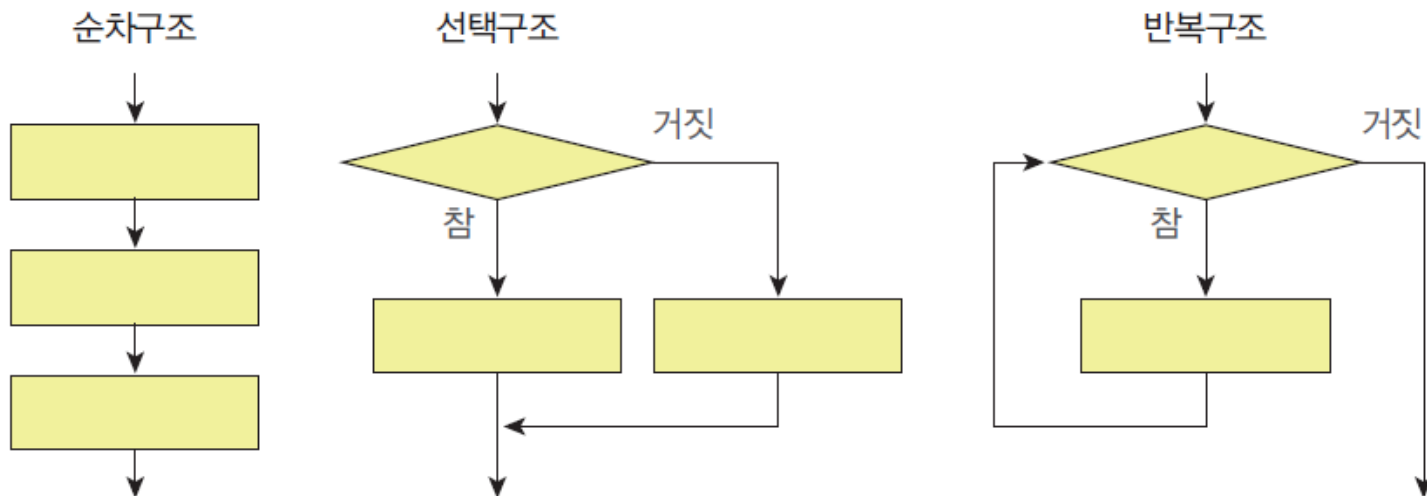


(2) 정수의 부호에 따라서 거북이를 (100, 100), (100, 0), (100,-100)으로 움직이는 프로그램을 작성해보자.

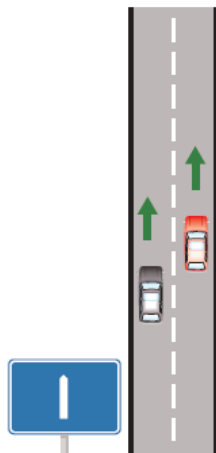


- 5.1 기본 제어구조

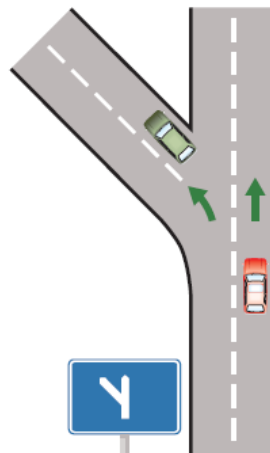
- 순차 구조(sequence) - 명령들이 순차적으로 실행되는 구조이다.
- 선택 구조(selection) - 둘 중의 하나의 명령을 선택하여 실행되는 구조이다.
- 반복 구조(iteration) - 동일한 명령이 반복되면서 실행되는 구조이다.



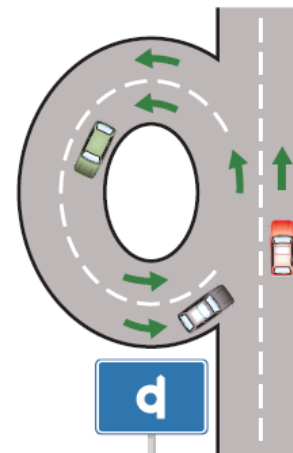
- 프로그램의 기본 블록을 쉽게 이해하려면 이것을 자동차(CPU)가 주행하는 도로로 생각하면 된다.



순차 구조



선택 구조



반복 구조

- 선택 구조가 없다면 프로그램은 항상 동일한 동작만을 되풀이할 것이다.
- (예) 자율 주행 자동차 프로그램이 신호등이나 전방 장애물에 따라서 동작을 다르게 하지 않는다면 어떻게 될까?



■ 부울형은 True와 False 값을 가짐

- '참'과 '거짓'을 표현하는데 사용

True	'참'을 의미
False	'거짓'을 의미

```
>>> a = True
>>> type(a)
<class 'bool'>
>>> a
True
```

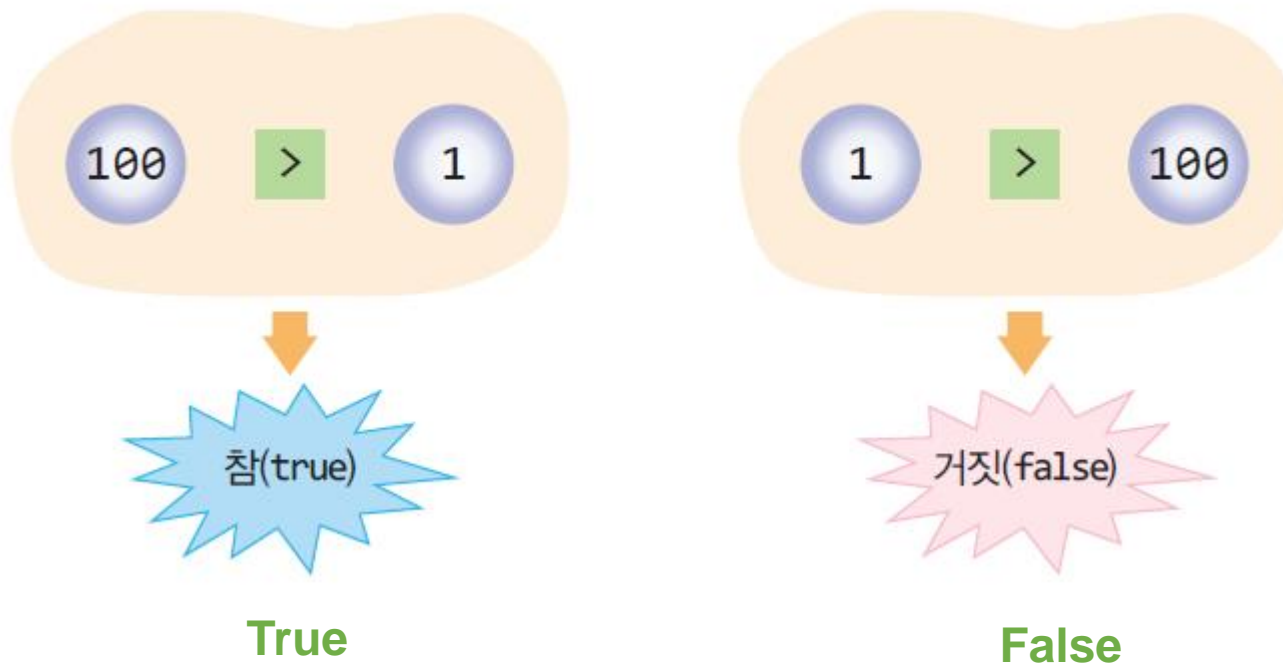
```
>>> a = TRUE
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in
    <module>
    a = TRUE
NameError: name 'TRUE' is not
defined
```

```
>>> a = False
>>> b = a
>>> type(b)
<class 'bool'>
>>> b
False
```

- **관계 연산자(relational operator)는 두 개의 피연산자를 비교하는 연산자**
 - 결과로 부울 값을 제공

x == y	x와 y의 값이 같으면 True를 반환하는 연산자
x != y	x와 y의 값이 다르면 True를 반환하는 연산자
x > y	x의 값이 y의 값보다 크면 True를 반환하는 연산자
x < y	x의 값이 y의 값보다 작으면 True를 반환하는 연산자
x >= y	x의 값이 y의 값보다 크거나 같으면 True를 반환하는 연산자
x <= y	x의 값이 y의 값보다 작거나 같으면 True를 반환하는 연산자

- 관계 수식은 참(True)이나 거짓(False)을 생성한다.



숫자를 입력 받아서 2의 배수인지의 여부를 flag에 할당해보자.

```
>>> n = input("Enter a number : ")
```

```
Enter a number : 10
```

```
>>> flag = int(n) % 2
```

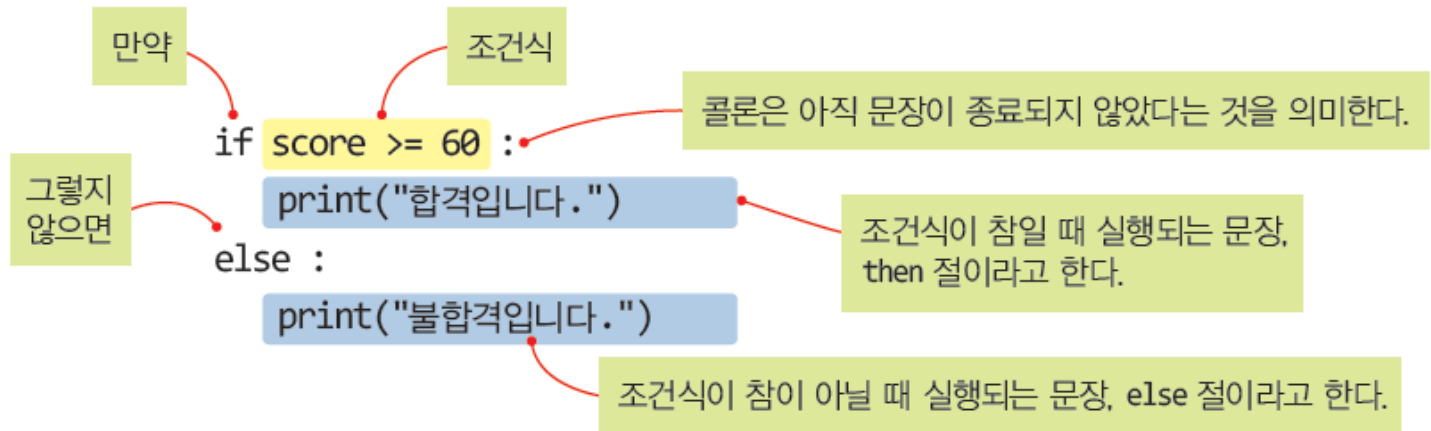
```
>>> flag == 0
```

```
True
```

- 5.2 if-else 문

파이썬에서 선택 구조를 위해 if-else문을 사용

if-else 문



```
score = int(input("성적을 입력하시오: "))  
  
if score >= 60:  
    print("합격입니다.")  
else:  
    print("불합격입니다.")
```

성적을 입력하시오: 80
합격입니다.

```
num = int(input("정수를 입력하시오: "))  
  
if num % 2 == 0 :  
    print("짝수입니다.")  
else:  
    print("홀수입니다.")
```

정수를 입력하시오: 10
짝수입니다.

- 만약 조건이 참인 경우에 여러 개의 문장이 실행되어야 한다면 어떻게 해야 하는가?

블록문

```
if score > 90 :
```

```
    print("합격입니다.")
```

```
    print("장학금도 받을 수 있습니다.")
```

블록: 여러 문장들을 묶은 것이다.

Lab2: 영화 나이 제한 검사



나이를 입력하시오: 19
이 영화를 보실 수 있습니다.

나이를 입력하시오: 14
이 영화를 보실 수 없습니다.




```
age = int(input("나이를 입력하시오: "))  
  
if age >= 15:  
    print("이 영화를 보실 수 있습니다.")  
else:  
    print("이 영화를 보실 수 없습니다.")
```



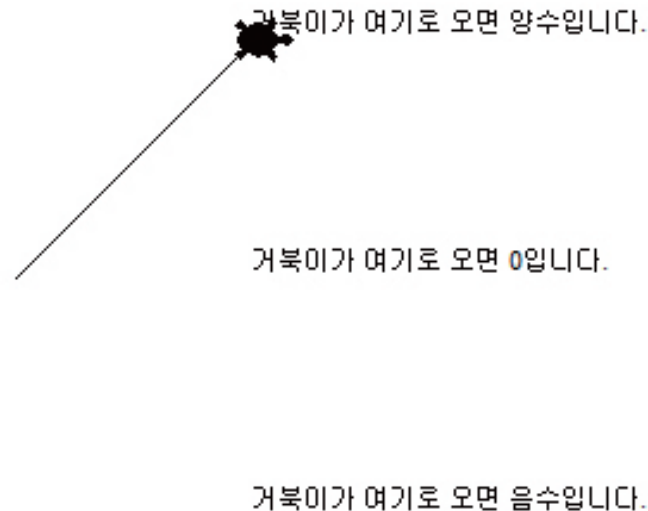
도전문제

15세 이상이면 “이 영화를 보실 수 있습니다.” 메시지에 추가로 “영화의 가격은 10000원입니다.”를 출력해보자. 만약 15세 미만이면 “이 영화를 보실 수 없습니다.” 메시지에 추가로 “다른 영화를 보시겠어요?”를 출력해보자.

Lab1: 부호에 따라 거북이를 움직이자



- 사용자로부터 정수를 받아서 정수의 부호에 따라서 거북이를 (100, 100), (100, 0), (100,-100)으로 움직이는 프로그램을 작성해보자.



```
import turtle
t = turtle.Turtle()
t.shape("turtle")

t.penup()                                     # 펜을 올려서 그림이 그려지지 않게 한다.
t.goto(100, 100)                             # 거북이를 (100, 100)으로 이동시킨다.
t.write("거북이가 여기로 오면 양수입니다.")
t.goto(100, 0)
t.write("거북이가 여기로 오면 0입니다.")
t.goto(100, -100)
t.write("거북이가 여기로 오면 음수입니다.")

t.goto(0, 0)                                # (0, 0) 위치로 거북이를 이동시킨다.
t.pendown()                                  # 펜을 내려서 그림이 그려지게 한다.
s = turtle.textinput("", "숫자를 입력하시오: ")
n=int(s)
if( n > 0 ):
    t.goto(100, 100)
if( n == 0 ):
    t.goto(100, 0)
if( n < 0 ):
    t.goto(100, -100)
```

조건 1

조건 2

나이가 10살 이상이고, 그리고 키가 165cm 이상이면
→ 놀이기구를 탈 수 있다.

(나이가 10살 이상이다) and (키가 165cm 이상이다)
→ 놀이기구를 탈 수 있다.

그리고

age \geq 10 and height \geq 165

나이가 10살 이상

키가 165cm 이상



■ 부울형 값들의 연산자

- 조건문이 하나 이상의 조건들을 포함하는 경우에 사용

■ '그리고' : AND

■ '또는' : OR

■ '부정' : NOT

■ 예

- 외국어 선생님을 뽑는 광고 : '영어와 중국어 능력 시험 성적이 1등급인 강사를 모집합니다'
- '영어 능력 == 1등급' 과 '중국어 능력 == 1등급'
- 두 조건 모두 True 인 경우에만 True

■ 불리언 식 (Boolean Expression)

- 부울 값 과 논리 연산자로 이루어진 식
- 결과로 부울형 반환

연산	의미
x and y	AND 연산, x와 y가 모두 참이면 참, 그렇지 않으면 거짓
x or y	OR 연산, x나 y중에서 하나만 참이면 참, 모두 거짓이면 거짓
not x	NOT 연산, x가 참이면 거짓, x가 거짓이면 참

- 불리언 식의 결과는 집합의 진리표(Truth Table)와 같음

p	q	$p \text{ and } q$	$p \text{ or } q$	$\text{not } p$	$\text{not } q$
False	False	False	False	True	True
False	True	False	True	True	False
True	False	False	True	False	True
True	True	True	True	False	False

- 다음 조건에 따라 부울 식을 작성하고 결과를 확인해보자.

변수 a = True이고 b = False이다.

▶ 'a and b'의 결과를 확인한다.

▶ 'a or b'의 결과를 확인한다.

▶ 'not b'의 결과를 확인한다.

```
>>> a = True
>>> b = False
>>> a and b
False
>>> a or b
True
>>> not b
True
```


a = 26, b = 31, c = 54 인 경우

1) (a > b) and (a < c) 의 결과는?

```
>>> a = 26
>>> b = 31
>>> c = 54
>>> (a > b) and (a < c)
False
```

■ 요구사항

어느 놀이 공원에서는 여자 신입생에게는 20% 할인 혜택을 제공한다고 한다.

▶ 수지는 여자이고 2학년이다. 수지가 할인 혜택을 받을 수 있는 지 부울 식을 사용하여 확인해보자.

(여성은 female, 남성은 male로 한다.)

수지 = '여자'
수지의 학년 = 2
만약 (수지의 성별 == '여자') and (수지의 학년 == 1)
수지는 할인을 받는다.

```
>>> std_gender = 'female'
>>> grade = 2
>>> (std_gender == 'female') and (grade == 1)
False
>>>
```

■ 요구사항 및 문제 해결 알고리즘

어느 회사의 입사 조건은 토익 점수가 800점 이상이거나 교양 영어에서 A학점이 만족되어야 지원 가능하다고 한다.

길동이는 토익이 900점이고 교양 영어에서 B학점이다. 길동이가 이 회사에 지원이 가능한지 부울 식을 사용하여 확인해보자.

길동의 토익 점수 = 900
길동의 영어 학점 = 'B'
만약 (길동의 토익 점수 >= 800) or (길동의 영어 학점 == 'A')
길동은 회사에 지원할 수 있다.

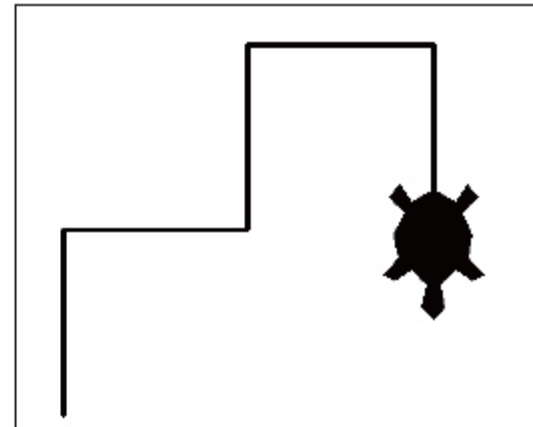
```
>>> toeic_score = 900
>>> grade_Eng = 'B'
>>> (toeic_score >= 800) or (grade_Eng == 'A')
True
>>>
```

Lab3: 거북이 제어하기



- 파이썬 셸에서 "l"을 입력하면 거북이가 왼쪽으로 100픽셀 이동하고 "r"을 입력하면 거북이가 오른쪽으로 100픽셀 이동하는 프로그램을 작성해

```
*Python 3.5.1 Shell*
File Edit Shell Debug Options Window
=====
명령을 입력하시오: l
명령을 입력하시오: r
명령을 입력하시오: l
명령을 입력하시오: r
명령을 입력하시오: r
```



- 아직 학습하지 않았지만 다음과 같은 코드를 사용하면 무한 반복할 수 있다.

```
while True :
```

```
    ...  
    ...  
    ...
```

```
import turtle

# 거북이를 만든다.
t = turtle.Turtle()

# 커서의 모양을 거북이로 한다.
t.shape("turtle")

# 거북이가 그리는 선의 두께를 3으로 한다.
t.width(3)

# 거북이를 3배 확대한다.
t.shapesize(3, 3)

# 무한 루프이다.
while (True):
    command = input("명령을 입력하시오: ")
    if command == "l":
        t.left(90)
        t.forward(100)
    if command == "r":
        t.right(90)
        t.forward(100)
```



- 입력된 연도가 윤년인지 아닌지를 판단하는 프로그램을 만들어 보자.

연도를 입력하시오: 2012
2012 년은 윤년입니다.

- ✓ 연도가 4로 나누어 떨어지면 윤년이다.
- ✓ 100으로 나누어 떨어지는 연도는 제외한다.
- ✓ 400으로 나누어 떨어지는 연도는 윤년이다.



((year % 4 == 0) and (year % 100 != 0)) or (year % 400 == 0)

연도가 4로 나누어떨어진다.

100으로 나누어떨어지는 연도는 제외한다.

400으로 나누어떨어지는 연도는 윤년이다.



Lab4: Solution



```
year = int(input("연도를 입력하시오: "))  
  
if ( (year % 4 == 0 and year % 100 != 0) or year % 400 == 0):  
    print(year, "년은 윤년입니다.")  
else :  
    print(year, "년은 윤년이 아닙니다.")
```

FEBRUARY 2012						
SUN	MON	TUE	WED	THU	FRI	SAT
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29			

Lab5: 동전 던지기 게임



- 동전을 던지기 게임을 작성해 보자.
- `import random`한 후에 `random.randrange(2)` 하면 0이나 1을 랜덤하게 생성할 수 있다.

동전 던지기 게임을 시작합니다.
뒷면입니다.
게임이 종료되었습니다.



```
import random

print("동전 던지기 게임을 시작합니다.")
coin = random.randrange(2)

if coin == 0 :
    print("앞면입니다.")
else :
    print("뒷면입니다.")
print("게임이 종료되었습니다.")
```



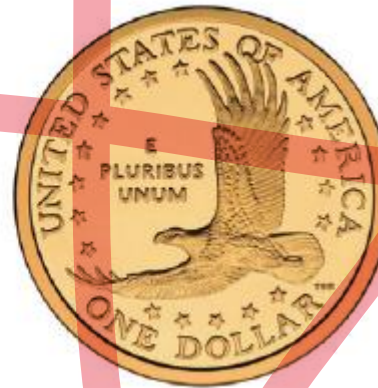
Lab6: 동전 던지기 게임(그래픽 버전)



- 동전을 던지기 게임을 그래픽 버전으로 만들어보자.



또는



```
screen = turtle.Screen() # 이미지를 사용하기 위해 터틀 그래픽 생성
image1 = "d:\\front.gif " # 이미지를 추가한다.
image2 = "d:\\back.gif"
screen.addshape(image1) # 이미지를 추가한다.
screen.addshape(image2) # 이미지를 추가한다.
t1.shape(image1) # 거북이의 모양을 설정한다.
t1.stamp() # 현재 위치에 거북이를 찍는다.
```

```
import turtle
import random # 난수 모듈을 불러온다.

screen = turtle.Screen()
image1 = "front.gif"
image2 = "back.gif"
screen.addshape(image1) # 이미지를 추가한다.
screen.addshape(image2) # 이미지를 추가한다.

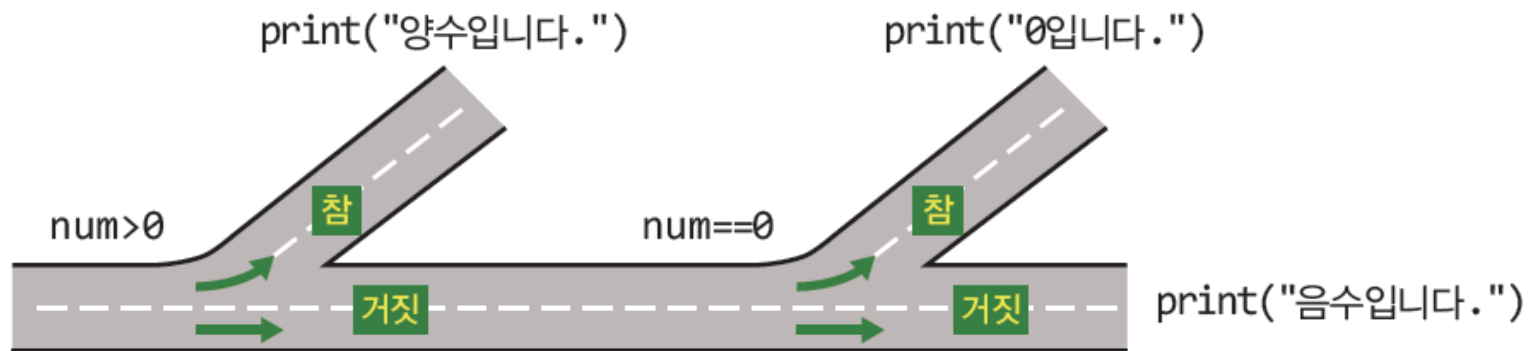
t1 = turtle.Turtle() # 첫 번째 거북이를 생성한다.

coin = random.randint(0, 1)

if coin == 0 :
    t1.shape(image1) # 거북이의 모양을 설정한다.
    t1.stamp() # 현재 위치에 거북이를 찍는다.
else :
    t1.shape(image2)
    t1.stamp()
```

- 5.3 if-elif-else 문

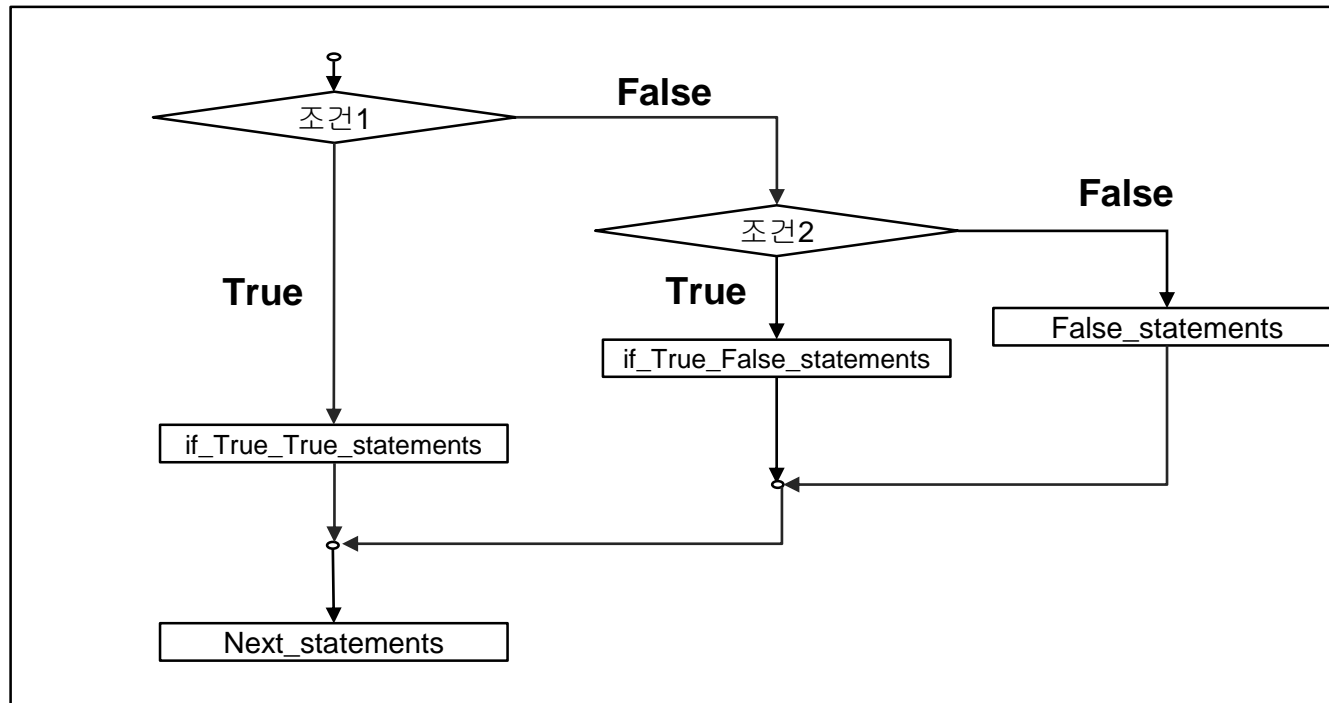
- 다음과 같이 진행하는 코드를 작성하려면?



if – elif – else 조건문 (1/3)

```
if 조건1 :  
    True_statements_a  
elif 조건2 :  
    True_statements_b  
else :  
    False_statements  
Next_statements
```

"조건1"이 True 이면 True_statements_a를 실행한 후, Next_statements를 실행
"조건1"이 False 이고 "조건2"가 True 이면 True_statements_b를 실행,
"조건1"이 False 이고 "조건2"도 False 이면 False_statements를 실행한 후
Next_statements를 실행



▪ if - else문의 header 구성

if a > 20 :
변수 -----
비교 연산자 (==, !=, >, >=, <, <=)
조건 끝
값

elif a < 20 :
변수 -----
비교 연산자 (==, !=, >, >=, <, <=)
조건 끝
값

else :
조건 끝

1~8까지 입력 받은 숫자를 판별하는 프로그램을 작성해보자 (비교)

```
num = int(input('숫자: '))
```

```
if num < 5 :  
    if num < 3 :  
        if num < 2 :  
            print('1')  
        else:  
            print('2')  
    else :  
        if num < 4 :  
            print('3')  
        else :  
            print('4')
```

3번의 비교연산

조건이 수가 많아질 경우
프로그램 실행 시간이
→ **짧아진다**

```
else :  
    if num < 7 :  
        if num < 6 :  
            print('5')  
        else :  
            print('6')  
    else :  
        if num < 8 :  
            print('7')  
        else :  
            print('8')
```

But **가독성이 낮다**

```
num = int(input('숫자: '))
```

```
if num == 1 :  
    print('1')  
  
elif num == 2 :  
    print('2')  
  
elif num == 3 :  
    print('3')  
  
elif num == 4 :  
    print('4')
```

8번의 비교연산

조건이 수가 많아질 경우
프로그램 실행시간이
→ **길어진다**

```
elif num == 5 :  
    print('5')  
  
elif num == 6 :  
    print('6')  
  
elif num == 7 :  
    print('7')  
  
elif num == 8 :  
    print('8')
```

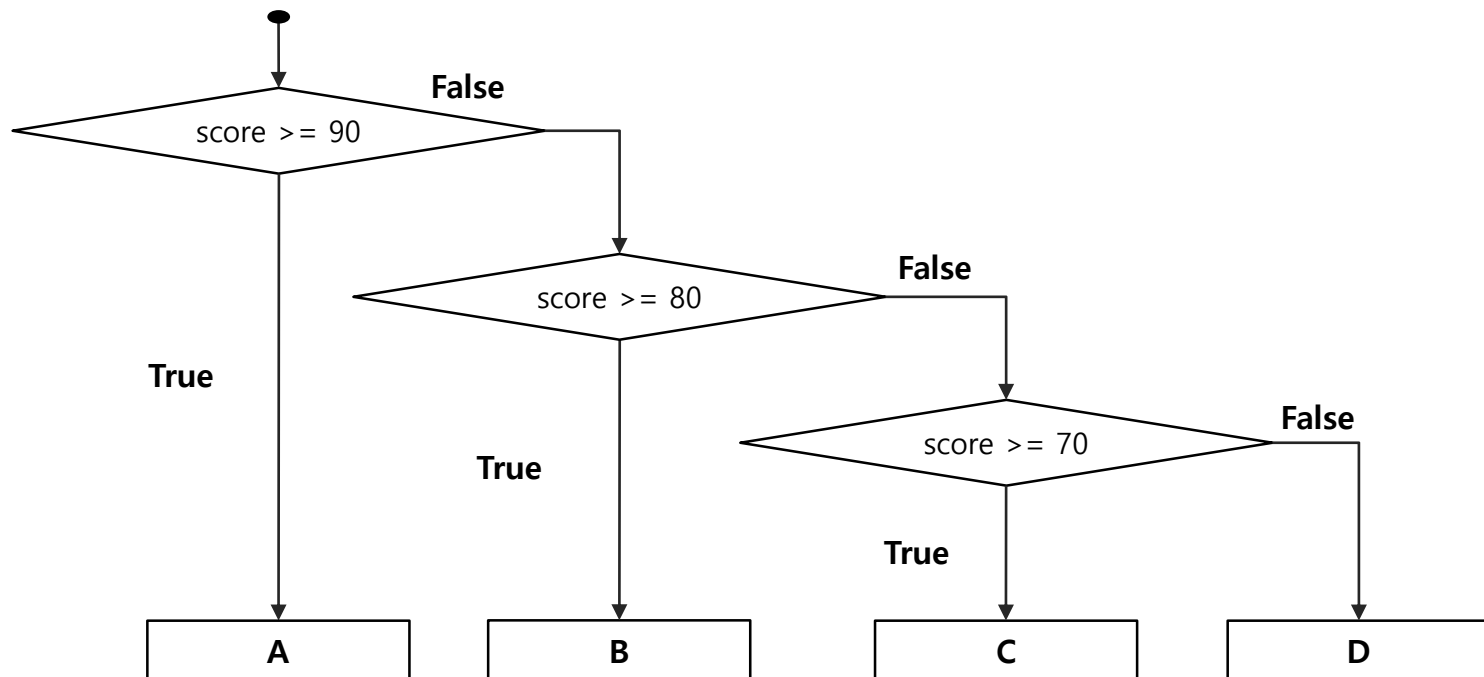
But **가독성이 높다**

[Test 5-5] if-elif-else문의 활용1 (1/2)



성적을 처리하여, 학점을 계산하려고 한다.
총점 90점 이상은 'A'학점을,
90미만 80점 이상은 'B'학점을,
80미만 70점 이상은 'C'학점을,
그 미만은 D학점을 주려고 한다.
총점을 입력 받아 성적을 출력하는 프로그램을 작성해보자.

■ 문제 해결 알고리즘



[Test 5-5] if-elif-else문의 활용1 (2/2)



- 프로그램

```
score = int(input('점수 입력 :'))

if score >= 90 :
    print('학점은 A')
elif score >= 80 :
    print('학점은 B')
elif score >= 70 :
    print('학점은 C')
else :
    print('학점은 D')
```

- 테스트

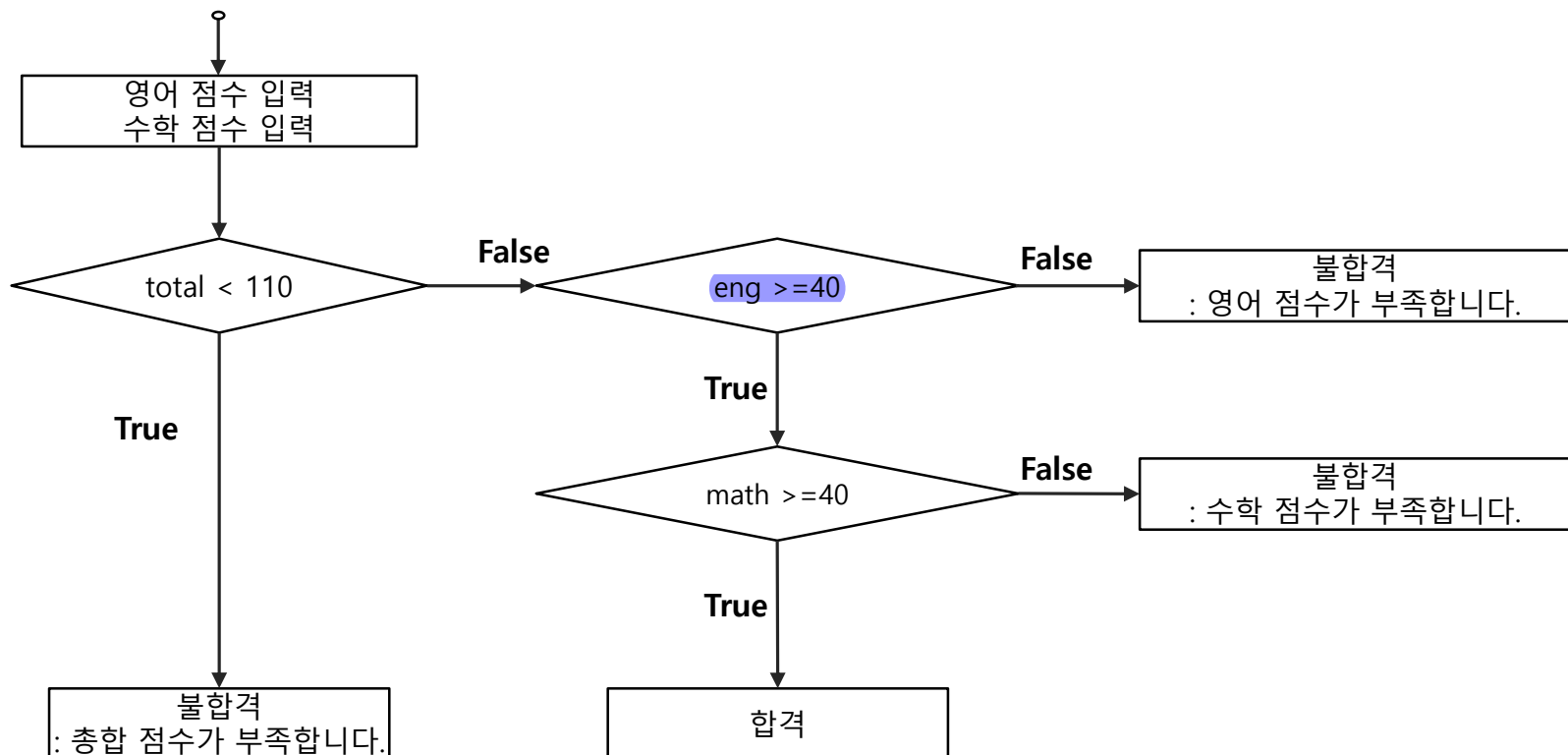
```
>>>
점수 입력 : 90
A
>>>
점수 입력 : 40
D
>>>
```

[Test 5-6] 학생수준평가 합격여부 출력 프로그램 (1/3)



학생수준평가 시험에서 영어 점수와 수학 점수가 합해서 **110점이 넘으면 합격**이지만 **각 점수가 40점 미만이면 불합격**이다.
영어(eng), 수학(math) 점수를 입력 받아 합격여부를 출력하는 프로그램을 작성해보자.

■ 문제 해결 알고리즘



- 프로그램

```
eng= int(input('영어 점수 입력 : '))

math= int(input('수학 점수 입력 : '))

total = eng + math

if total < 110 :
    print('불합격 : 총합 점수가 부족합니다')
elif eng >=40 and math >= 40 :
    print('합격')
elif eng >=40 and math < 40 :
    print('불합격 : 수학 점수가 부족합니다')
else:
    print('불합격 : 영어 점수가 부족합니다')
```

● 테스트

>>>

영어 점수 입력 : 80

수학 점수 입력 : 20

불합격 : 총합 점수가 부족합니다

>>>

영어 점수 입력 : 90

수학 점수 입력 : 30

불합격 : 수학 점수가 부족합니다

>>>

영어 점수 입력 : 35

수학 점수 입력 : 95

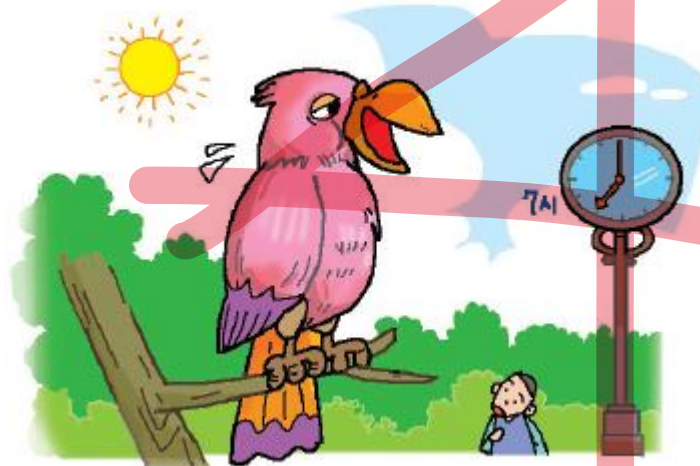
불합격 : 영어 점수가 부족합니다

>>>

Lab7: 종달새가 노래할까?



- 동물원에 있는 종달새가 다음과 같은 2가지 조건이 충족될 때 노래를 한다고 하자.
 - 오전 6시부터 오전 9시 사이
 - 날씨가 화창하다.



- 현재 시각을 난수로 생성하고 날씨도 [True, False] 중에서 랜덤하게 선택하자.
종달새가 노래를 부를 것인지, 조용히 있을 것인지를 판단해보자.

```
import random  
time = random.randint(1, 24)  
sunny = random.choice([True, False])
```

좋은 아침입니다. 지금 시각은 1시 입니다.
현재 날씨가 화창하지 않습니다.
종달새가 노래를 하지 않는다.

```
import random

time = random.randint(1, 24)
print("좋은 아침입니다. 지금 시각은 " + str(time) + "시 입니다.")

sunny = random.choice([True, False])

if sunny:
    print ("현재 날씨가 화창합니다. ")
else:
    print ("현재 날씨가 화창하지 않습니다. ")

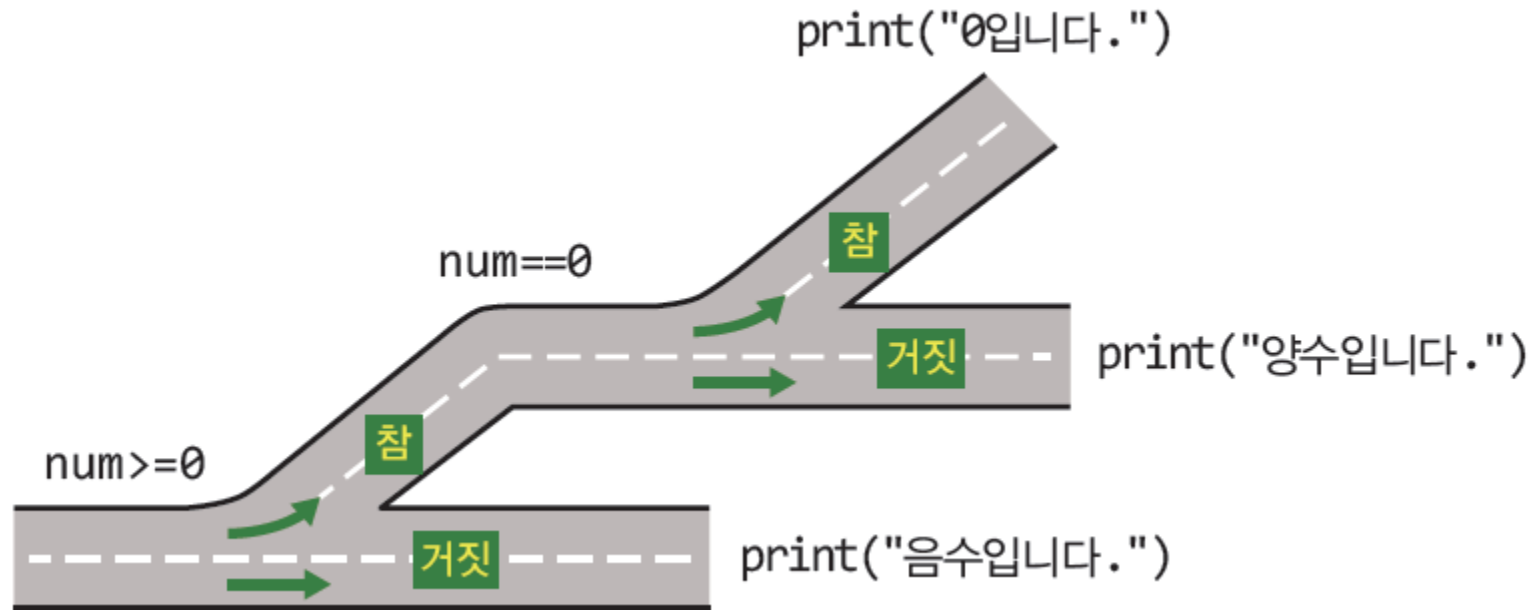
# 종달새가 노래를 할 것인지를 판단해보자.
if time >= 6 and time < 9 and sunny:
    print ("종달새가 노래를 한다.")
else :
    print ("종달새가 노래를 하지 않는다.")
```



도전문제

종달새가 6시와 9시 사이 또는 14시와 16시 사이에 노래한다고 하면(그리고 날씨는 항상 화창하여야 한다) 위 조건식을 어떻게 변경하여야 하는가?

- if 문 안에 다른 if 문이 들어갈 수도 있다. 이것을 중첩 if 문이라고 한다.



- 앞의 [Test 5-6]의 예제를 중첩 if-else문으로

```
total = eng + math
if total < 110 :
    print('불합격 : 총합 점수가 부족합니다')
elif eng >= 40 :
    if math >= 40 :
        print('합격!!')
    else :
        print('불합격 : 수학 점수가 부족합니다')
else:
    print('불합격 : 영어 점수가 부족합니다')
```

>>>

영어 점수 입력 : 90

수학 점수 입력 : 30

불합격 : 수학 점수가 부족합니다



- 사용자로부터 아이디를 받아서 프로그램에 저장된 아이디와 일치하는지 여부를 출력하는 프로그램을 작성해보자.

아이디를 입력하시오: ilovepython
환영합니다.

아이디를 입력하시오: iloveruby
아이디를 찾을 수 없습니다.

```
id = "ilovepython"
s = input("아이디를 입력하시오: ")
if s == id:
    print("환영합니다.")
else:
    print("아이디를 찾을 수 없습니다.")
```



도전문제

아이디 검사가 종료되면 바로 패스워드 검사를 해보자. 즉 다음과 같은 출력을 가지는 프로그램을 작성한다.

```
아이디를 입력하시오: ilovepython
패스워드를 입력하시오: 123456
환영합니다.
```



- 사용자가 컴퓨터를 상대로 패널티 킥을 하는 축구게임을 만든다. 사용자는 3가지 영역 중 하나를 선택하여 패널티 킥을 하고, 컴퓨터도 난수를 생성하여 3개의 영역 중 하나를 수비한다.

어디를 공격하시겠어요?(왼쪽, 중앙, 오른쪽) 중앙
패널티 킥을 성공하였습니다.



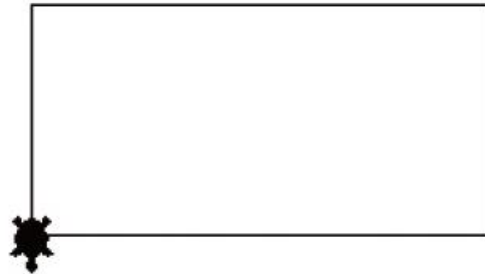
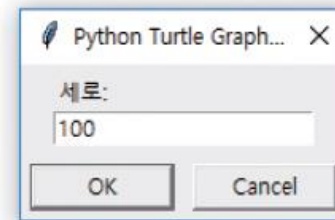
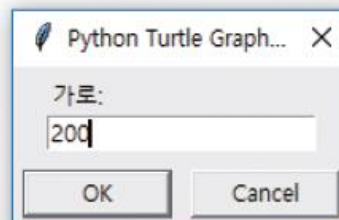
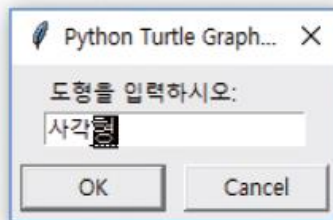

```
options=["왼쪽", "중앙", "오른쪽"]
computer_choice = random.choice(options)
user_choice = input("어디를 공격하시겠어요?(왼쪽, 중앙, 오른쪽) ")

if computer_choice == user_choice:
    print("컴퓨터가 수비에 성공하였습니다. ")
else:
    print("패널티 킥을 성공하였습니다. ")
```

```
import random
r = random.randrange(3)
attack = [' ', ' ', ' ']
you = input("어디를 공격하시겠어요?(왼쪽 상단, 왼쪽 하단, 중앙, 오른쪽 상단, 오른쪽 하단) ")
if attack[r] == you :
    print("성공하였습니다. ")
else :
    print("실패하였습니다. ")
```



- 터틀 그래픽을 이용하여 사용자가 선택하는 도형을 화면에 그리는 프로그램을 작성해보자. 도형은 "사각형", "삼각형", "원" 중의 하나이다. 각 도형의 치수는 사용자에게 물어보도록 하자.



```
import turtle

t = turtle.Turtle()
t.shape("turtle")

s = turtle.textinput("", "도형을 입력하시오: ")

if s == "사각형" :
    s = turtle.textinput("", "가로: ")
    w=int(s)
    s = turtle.textinput("", "세로: ")
    h=int(s)
    t.forward(w)
    t.left(90)
    t.forward(h)
    t.left(90)
    t.forward(w)
    t.left(90)
    t.forward(h)
```



도전문제

위의 프로그램에서 “사각형”만을 지원하고 있다. “삼각형”, “원”인 경우에 도형을 그리는 코드를 추가하라.

- $>$, $<$, $=$ 와 같은 관계 연산자를 학습하였다.
- 논리 연산자 `and`나 `or` 를 사용하면 조건들을 묶을 수 있다.
- 블록은 조건이 맞았을 때 묶어서 실행되는 코드로 파이썬에서 **들여쓰기로 블록을 만든다.**
- `if-else` 문 안에 다른 `if-else` 문이 포함될 수 있다.



Q & A

