

# 파이썬 프로그래밍 입문

비교문(조건문, 선택문)

최 윤 정

cris.lecture@gmail.com



일상생활의 대부분은 선택의 연속!

컴퓨터를 통해서 해결하려는 대부분의 목표들은 선택의 연속과정으로 생각할 수 있음

# 오늘은

---

## □ 비교문에 대해 알아봅니다.

- 한번 비교하기
- 여러 번 비교하기
- 복잡한 조건문은 어떻게 처리할까?

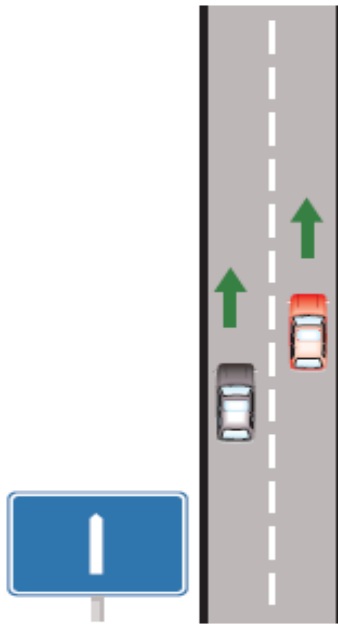
## □ 실습

- 거북이(선택)
- 로그인 시스템
- 성적으로 학점구하기
- 축구게임
- ...

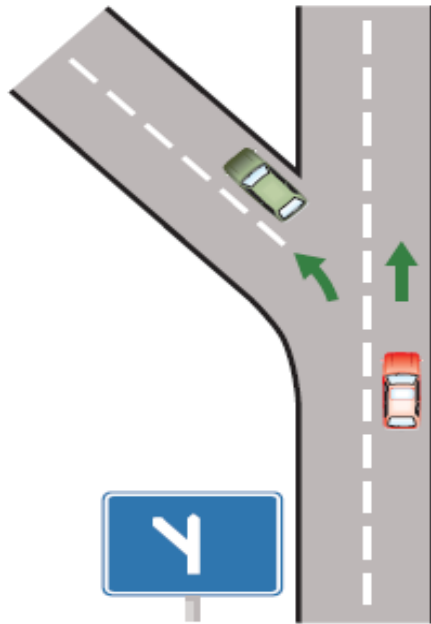
# 3가지의 기본 제어구조

## □ 제어문

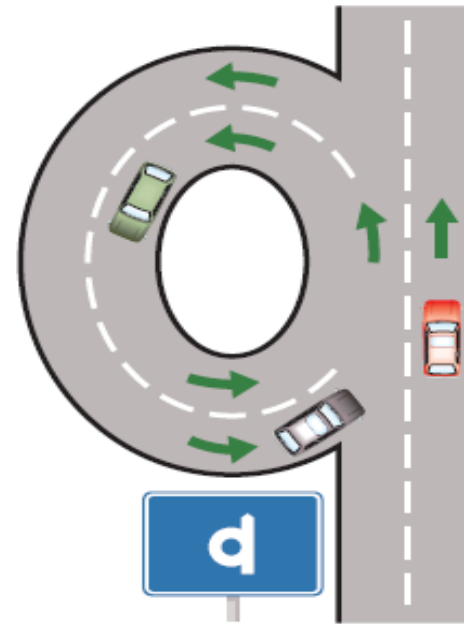
- 순차적으로 실행하는 명령문
- 조건에 따라 비교하는 비교문
- 그리고 조건없는 무한 반복문



순차 구조



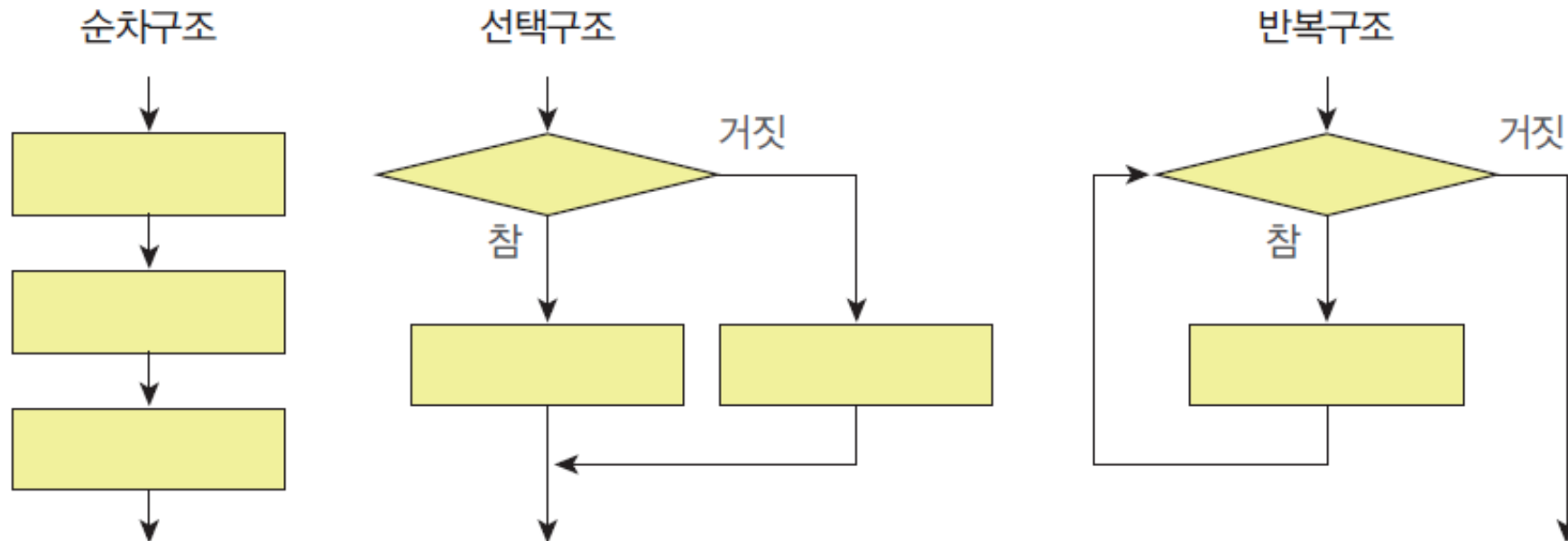
선택 구조



반복 구조

# 3가지의 기본 제어구조

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



# 비교 : 선택 구조가 필요한 이유

- 선택 구조가 없다면 프로그램은 항상 동일한 동작만 수행하게 된다.
- 교차로에 신호등이 없고, 음식점에 메뉴가 없다면 각각의 기능을 수행할 수 없다



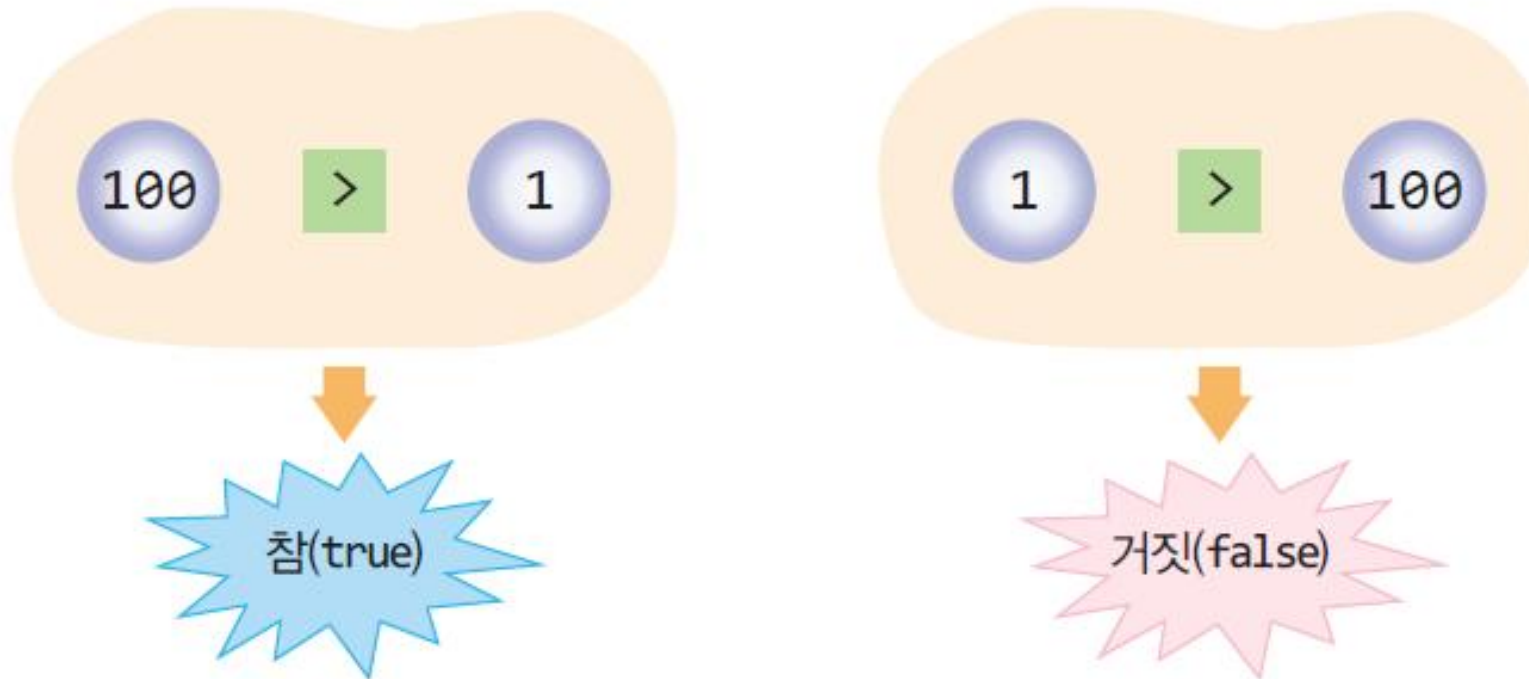
# 관계 연산자

- 산술연산자 : 더하기, 곱하기, 빼기, 나누기 등
- 관계 연산자(relational operator)
  - 비교에 사용하는 연산자
  - 두 개의 피연산자를 비교하는 연산자

연산	의미
$x == y$	x와 y가 같은가?
$x != y$	x와 y가 다른가?
$x > y$	x가 y보다 큰가?
$x < y$	x가 y보다 작은가?
$x >= y$	x가 y보다 크거나 같은가?
$x <= y$	x가 y보다 작거나 같은가?

# 관계 연산자의 결과값 : 참 또는 거짓

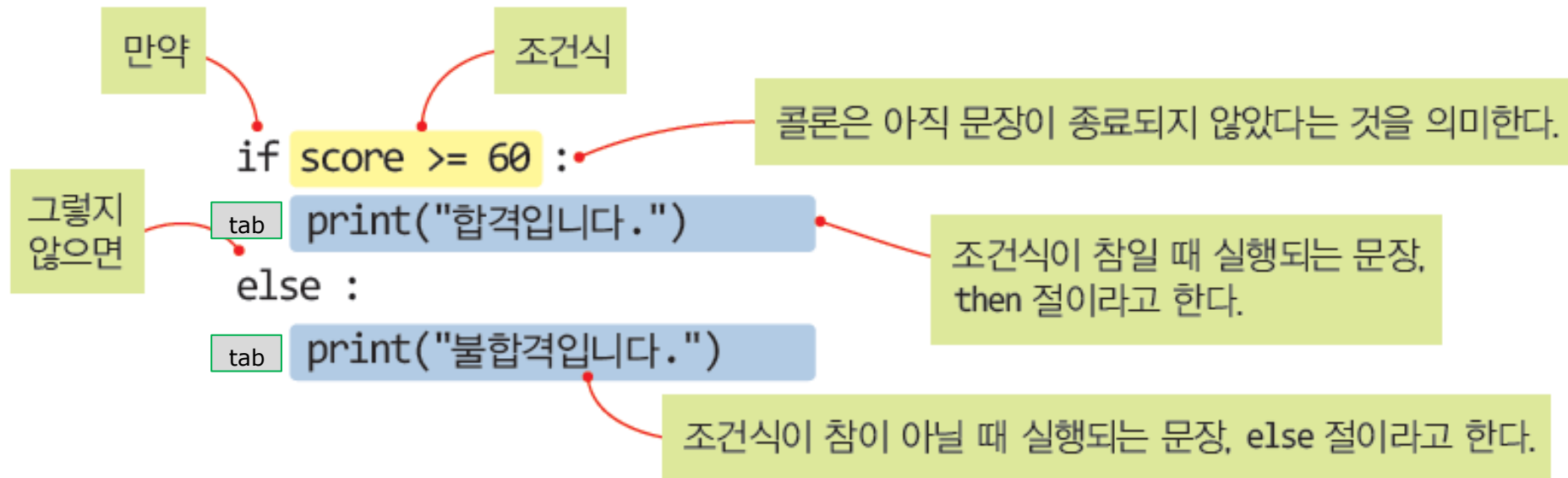
- 관계 수식은 참(True)이나 거짓(False)을 생성한다.
- 문제에 대한 정확한 답을 얻기 위해 주어진 문제로부터 명확한 조건을 추출하는 것이 가장 중요한 일





# 비교할 때 사용하는 문장 : if-else 문

## if-else 문



# 단순 if-else의 예 #1

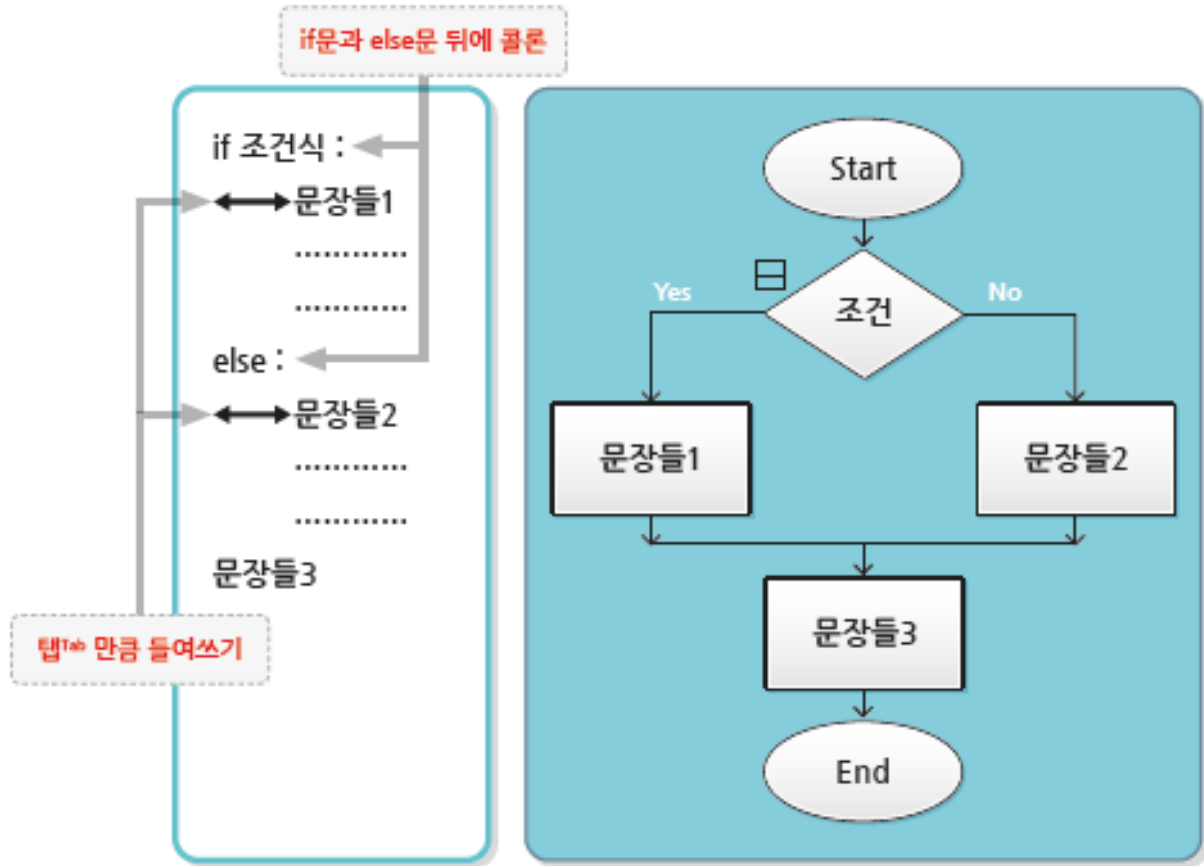


그림 4-5 이중 if문

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

성적을 입력하시오: 80  
합격입니다.

## 단순 if-else의 예 #2

```
num = int(input("정수를 입력하시오: "))

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

# 조건에 해당하는 실행문이 하나일 경우에는 붙여쓸 수 있다.

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

정수를 입력하시오: 10  
짝수입니다.

# 블록

## □ 만약 조건이 참인 경우에 여러 개의 문장이 실행되어야 한다면?

- 파이썬에서는 탭으로 블록을 구분하여 입력합니다
- 참고 : 다른 언어에서는 { } 를 사용합니다

블록문

```
if score > 90 :
```

```
    print("합격입니다.")
```

```
    print("장학금도 받을 수 있습니다.")
```

블록: 여러 문장들을 묶은 것이다.

# Lab: 영화 나이 제한 검사

나이를 입력하시오: 19  
이 영화를 보실 수 있습니다.

나이를 입력하시오: 14  
이 영화를 보실 수 없습니다.

```
limit = 15
```

```
print("%d세 이상만 관람이 가능한 영화입니다."%limit)
```

```
age = int(input("나이를 입력하시오: "))
```

```
if age >= limit:
```

```
    print("이 영화를 보실 수 있습니다.")
```

```
else:
```

```
    print("이 영화를 보실 수 없습니다.")
```



# 논리 연산자

조건 1

조건 2

나이가 10살 이상이고, 그리고 키가 140 cm 이상이면  
→ 놀이기구를 탈 수 있다.

(나이가 10살 이상이다) **and** (키가 140 cm 이상이다)  
→ 놀이기구를 탈 수 있다.

**(age >= 10) and (height >= 140)**



# 논리 연산자의 종류 : and, or, not

- 논리연산자와 관계연산자를 이용하여 주어진 문제로부터 명확한 조건을 추출하는 것이 프로그램의 핵심

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

# 잠깐 Quiz

## □ 다음 문장을 비교문으로 나타내보세요

- 나이가 18세 이상이고 25세 이하이면 원래 물건값을 30%할인해준다.
- 국어, 영어, 수학 성적이 모두 70점 이상이거나, 3과목 총합이 230점 이상이면 합격이다.
- 변수명은 자유롭게 사용하세요

```
물건값=10000
나이 = int(input("나이: "))
if (나이 >=18 and 나이 <=25) :
    물건값=물건값*0.7
#또는
#if (18 <=나이 <=25) : 물건값*=0.7
print(물건값)
```

```
국어=int(input("국어점수:"))
영어=int(input("영어점수:"))
수학=int(input("수학점수:"))
total= 국어+영어+수학
if( (국어>=70 and 영어>=70 and 수학>=70) or
    total >=230) :
    print("합격입니다.")
else : print("불합격입니다.")
```



# 복잡한 조건문의 예 :

If, elif, else문 뒤에 콜론  
각 문장들은 들여쓰기

```
if 조건식:
    ←문장들1
    .....
elif 조건식:
    문장들2
    ←.....
    .....
elif 조건식:
    문장들n
    .....
else:
    문장들n+1
    .....
```

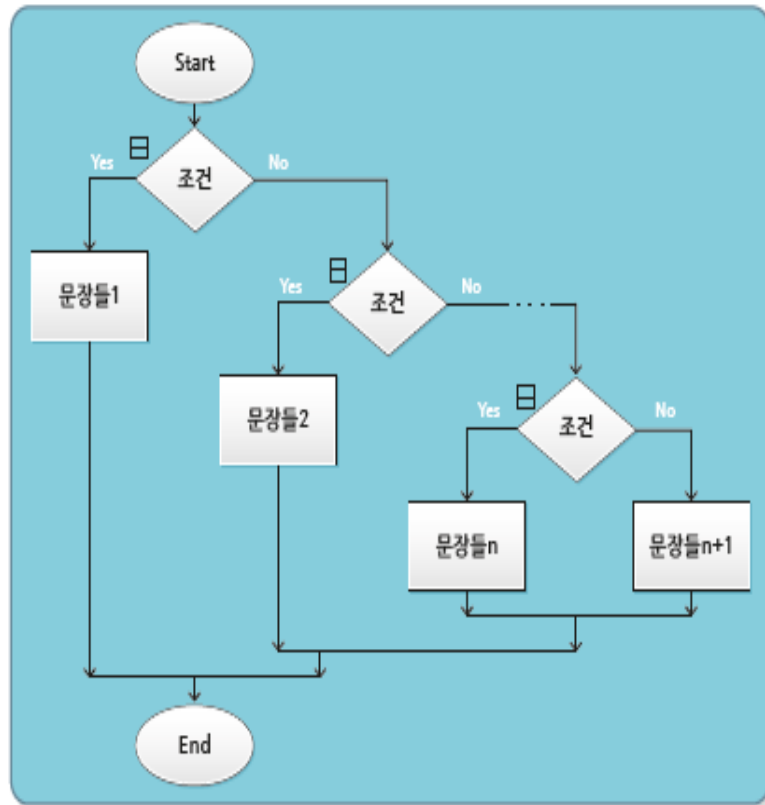


그림 4-6 다중 if문의 구조

if문내에 다른 if문이 내포되는 형태  
if와 쌍을 이루는 else문은 간격이 같아야 한다

```
if 조건식:
    if 조건식:
        ←문장들1
        .....
    else:
        ←문장들2
        .....
else:
    ←문장들3
    .....
else:
    ←문장들4
    .....
else:
    ←문장들5
    .....
```

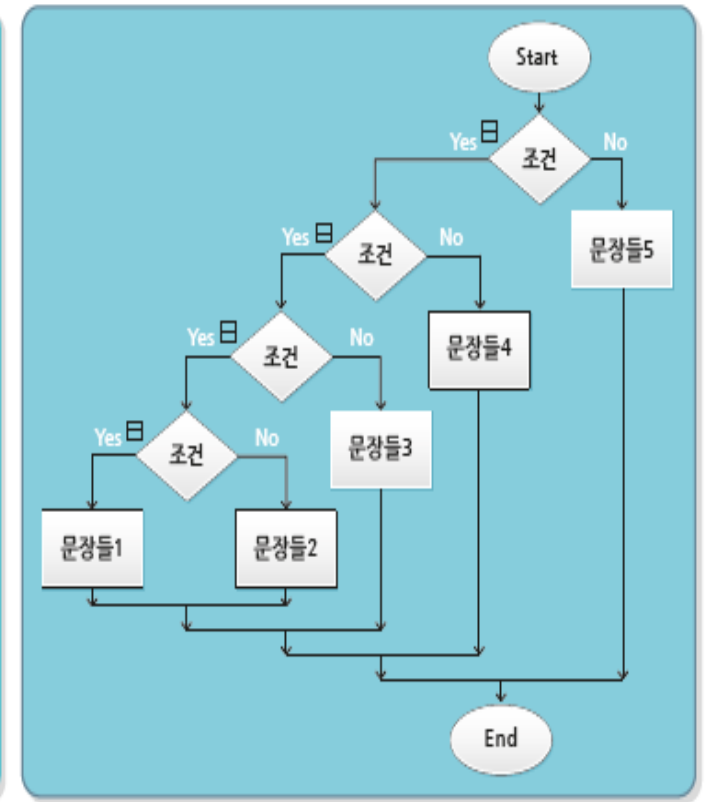
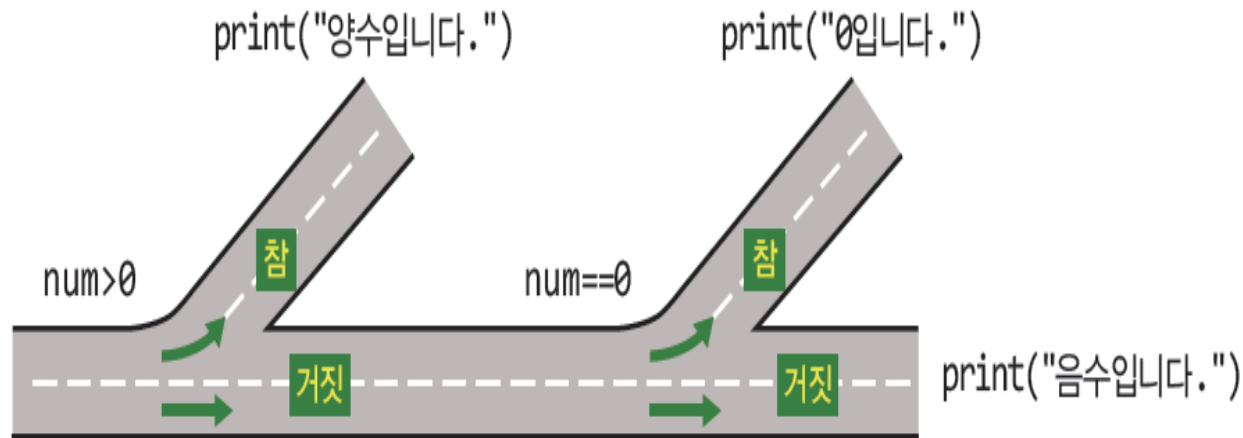


그림 4-7 중첩 if문의 구조

# 복잡한 조건문의 예 : if-elif-elif-....-else

□ 다음과 같이 진행하는 코드를 작성하려면?

- 만약에  $num > 0$ 이면 양수.
- 그렇지않고 만약에  $num$ 가 0이면 0.
- 그렇지않다면  $num$ 은 음수.
- **else if** → 줄여서 **elif**



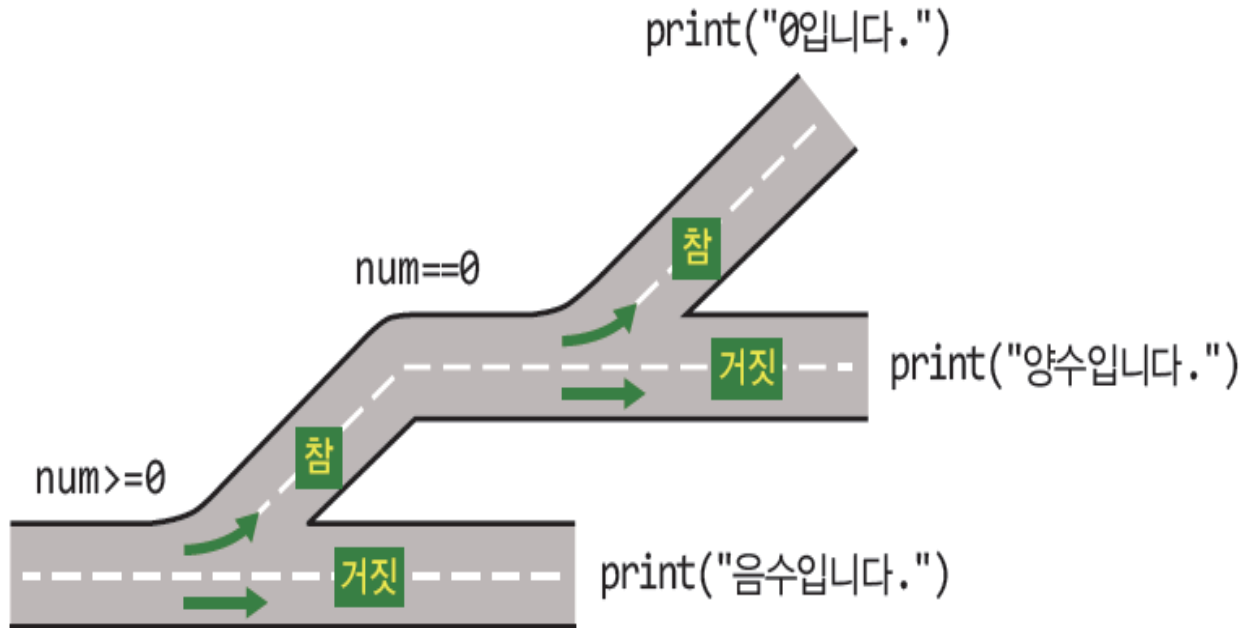
```
num = int(input("정수를 입력하시오: "))  
  
if num > 0:  
    print("양수입니다.")  
elif num == 0:  
    print("0입니다.")  
else:  
    print("음수입니다.")
```

정수를 입력하시오: 10  
양수입니다.

# 복잡한 조건문의 예 : 중첩 if-else문

□ if 문 안에 다른 if 문이 들어갈 수도 있다.

- 만약에  $num \geq 0$  인 경우
  - 만약에  $num == 0$ 이면 0
  - 그렇지 않으면 양수
- 그렇지않다면 num은 음수.



```
num = int(input("정수를 입력하시오: "))
```

```
if num >= 0:
    if num == 0:
        print("0입니다.")
    else:
        print("양수입니다.")
else:
    print("음수입니다.")
```

정수를 입력하시오: 10  
양수입니다.

# Lab: 세 개의 수를 입력받아 가장 큰 수 구하기

- 세개의 수를 입력 받은 후 가장 큰 수를 출력하세요
- 비교하는 방법은 여러가지가 있지만 주어진 문제로부터 명확하고 효율적인 조건을 추출하는 방법을 생각하는 것이 중요합니다!
- 실행 예)

```
PS D:\python코드> & C:/Users/user/AppData/Local/Programs/Python/Python38-32/python.exe D:\python코드\lab1.py
정수1:5
정수2:4
정수3:10
가장 큰 수는 10입니다.!
```

```
PS D:\python코드> & C:/Users/user/AppData/Local/Programs/Python/Python38-32/python.exe D:\python코드\lab1.py
정수1:10
정수2:2
정수3:9
가장 큰 수는 10입니다.!
```

```
PS D:\python코드> 
```

```
1  #3개의 정수 입력받기
2  a = int(input("정수1:"))
3  b = int(input("정수2:"))
4  c = int(input("정수3:"))
5
6  #a가 가장 크다고 가정
7  result = a
8  if result < b : result=b
9  if result < c : result=c
10
11 print(f"가장 큰 수는 {result}입니다.!")
12 
```

# Lab: 동전 던지기 게임

## □ 랜덤(난수) 수를 발생시켜 봅시다.

- import random
- 사용법 :
  - random.randrange(3) : 0,1,2 중 랜덤 수를 구한다
  - random.randint(0,100) : 0부터 100까지 랜덤 수
- 앞, 뒤 ➔ 1 또는 0으로 처리



```
import random    #난수 라이브러리를 이용하여

print( " 동전 던지기 게임을 시작합니다. " )
coin = random.randrange(2)
if coin == 0 : print("앞면입니다.")
else : print("뒷면입니다.")

print("게임이 종료되었습니다.")
```

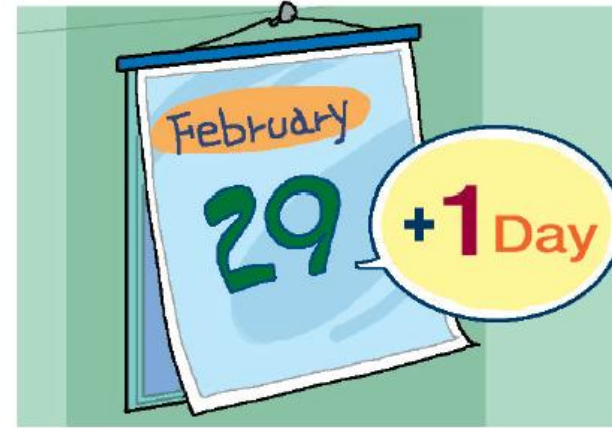
동전 던지기 게임을 시작합니다.  
뒷면입니다.  
게임이 종료되었습니다.

# Lab: 윤년 판단

- 입력된 연도가 윤년일까 판단해보자
- 1년이 365일이면 평년, 366일이면 윤년이다

연도를 입력하시오: 2012  
2012 년은 윤년입니다.

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



예) 2024, 2028, 2096은 윤년  
2100, 2200, 2300은 평년

# 윤년의 조건

$( (year \% 4 == 0) \text{ and } (year \% 100 != 0) ) \text{ or } (year \% 400 == 0)$

연도가 4로 나누어떨어진다.

100으로 나누어떨어지는 연도는 제외한다.

400으로 나누어떨어지는 연도는 윤년이다.

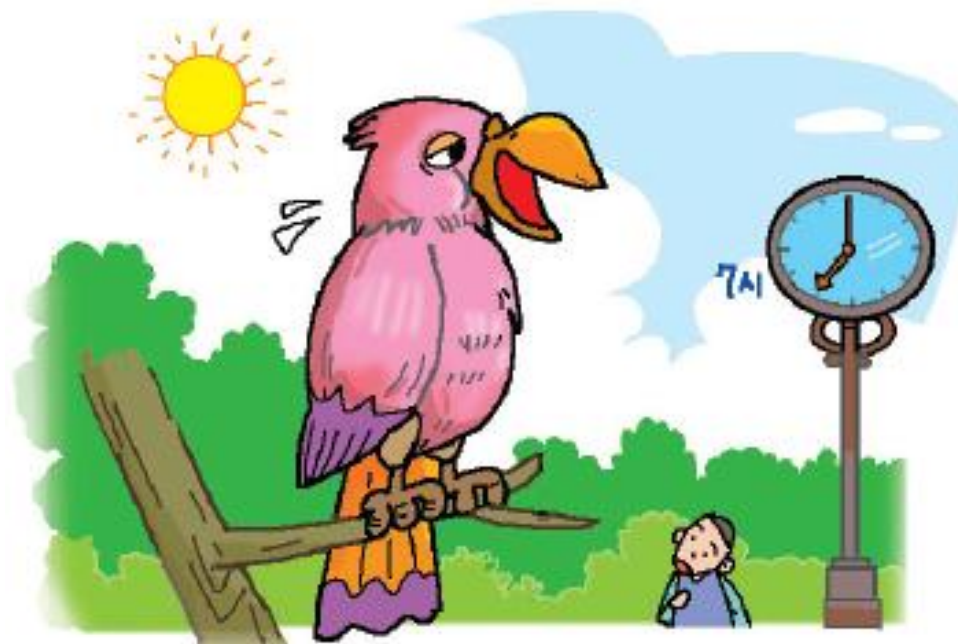
```
year = int(input("연도를 입력하시오: "))
if ( (year % 4 == 0 and year % 100 != 0) or year % 400 == 0 ):
    print(year, "년은 윤년입니다.")
else :
    print(year, "년은 윤년이 아닙니다.")
```

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			

# Lab: 종달새가 노래할 수 있을까?



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





# random 이용

## □ 현재 시각을 난수로 생성하고

- `random.randint(1,24)`

## □ 현재 날씨도 난수로 생성하자.

- 날씨가 화창하면 True, 그렇지 않으면 False로 하여 [True, False] 중에서 랜덤하게 선택하자.
- 둘 중 하나 고르기
  - `random.choice([True, False])`
  - `random.choice(['a','b'])`
- 숫자 범위에서 랜덤수 얻기
  - `random.randrange(3)` : 0,1,2 중 랜덤 수를 구한다
  - `random.randint(0,100)` : 0부터 100까지 랜덤 수

```
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.

# Solution

```
import random
time = random.randint(1, 24)
print(f"좋은 아침입니다. 지금 시각은 {time} 시 입니다.")
```

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

```
if sunny==True :
```

또는 if sunny :

```
    print ("현재 날씨가 화창합니다. ")
```

```
else:
```

```
    print ("현재 날씨가 화창하지 않습니다. ")
```

```
if time >= 6 and time < 9 and sunny:
```

또는 if 6<=time <9 and sunny==True

```
    print ("종달새가 노래를 한다.")
```

```
else :
```

```
    print ("종달새가 노래를 하지 않는다.")
```

# Lab : 가벼운 실습

---

Turtle : 거북이 방향전환

로그인 시스템

짱구의 성적표

축구게임

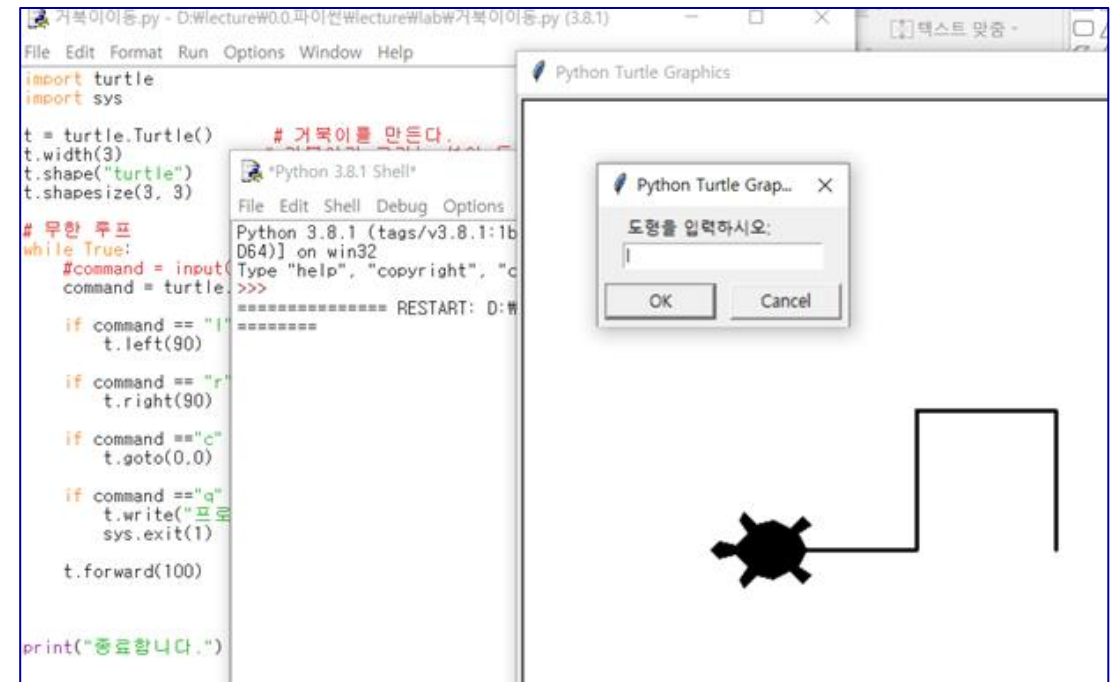
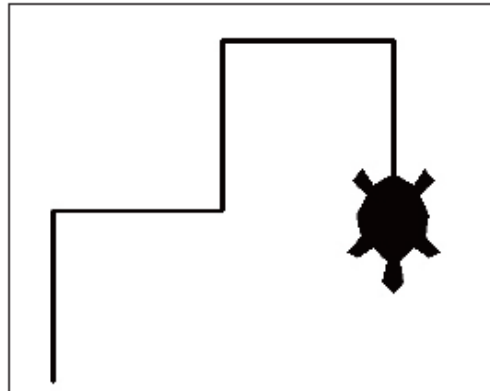
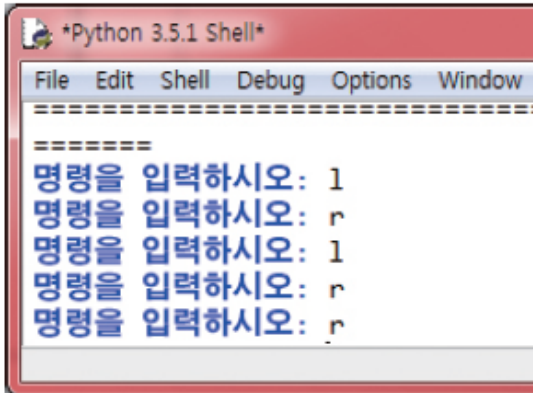
...

# Lab#1: 거북이 제어하기



## □ 파이썬 셸에서

- "|"을 입력하면 거북이가 왼쪽으로 100픽셀
- "r"을 입력하면 거북이가 오른쪽으로 100픽셀 이동하는 프로그램
- "q"를 입력하면 "종료합니다" 를 출력하고 그만
- 무한히 반복해보자



# 무한 반복 구조

---

- 아직 학습하지 않았지만 다음과 같은 코드를 사용하면 무한 반복!
- 들여쓰기로 무한 반복할 블록을 지정합니다.

```
while True:
```

```
    ...
```

```
    ...
```

```
    ...
```

# Solution

```
import turtle
t = turtle.Turtle()    # 거북이를 만든다.
t.width(3)             # 거북이가 그리는 선의 두께를 3으로 한다.
t.shape("turtle")      # 커서의 모양을 거북이로 한다.
t.shapesize(3, 3)      # 거북이를 3배 확대한다.
```

# 무한 루프

while True:

```
    command = input("명령을 입력하시오: ")
    if command == "l":    # 사용자가 "l"을 입력하였으면
        t.left(90)
    if command == "r":    # 사용자가 "r"을 입력하였으면
        t.right(90)

    t.forward(100)
```

# Solution - 응용

```
import turtle
import sys
```

```
t = turtle.Turtle()      # 거북이를 만든다.
t.width(3)               # 거북이가 그리는 선의 두께를 3으로 한다.
t.shape("turtle")        # 커서의 모양을 거북이로 한다.
t.shapesize(3, 3)        # 거북이를 3배 확대한다.
```

```
# 무한 루프
```

```
while True:
```

```
    # command = input("명령을 입력하시오: ")
    command = turtle.textinput("", "도형을 입력하시오: ")
```

```
    if command == "l" : # 사용자가 "l"을 입력하면 왼쪽으로 회
        t.left(90)
```

```
    if command == "r" : # 사용자가 "r"을 입력하면 오른쪽으로 회전
        t.right(90)
```

```
    if command == "c" : # 사용자가 "c"를 입력하면 원래 위치로 이동
        t.goto(0,0)
```

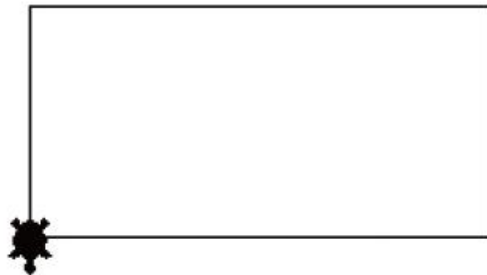
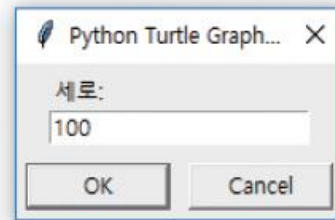
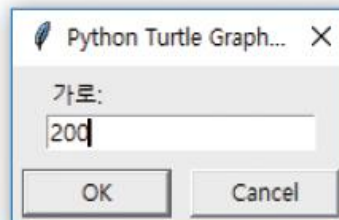
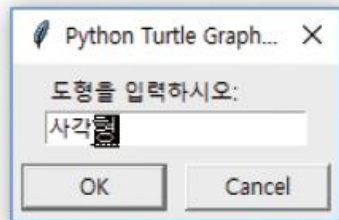
```
    if command == "q" :
        t.write("프로그램을 종료합니다.")
        sys.exit(1)
```

```
    t.forward(100)
```

```
print("종료합니다.")
```

## Lab#2: 도형그리기

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





# Solution

```
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)
```

#삼각형과 원은 여러분이 추가합니다.

## Lab#3: 로그인 프로그램

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

아이디를 입력하시오: ilovepython  
환영합니다.

아이디를 입력하시오: iloveruby  
아이디를 찾을 수 없습니다.

- 도전! 비밀번호도 받아보자!

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

아이디를 입력하시오: ilovepython  
패스워드를 입력하시오: 123456  
환영합니다.

# Solution

---

```
id = "ilovepython"
passwd="123456"
myID = input("아이디를 입력하시오: ")
if id ==myID :
    print("환영합니다.")
else:
    print("아이디를 찾을 수 없습니다.")
```

## Lab#4: 짱구의 성적표

---

### □ 성적을 입력받아서 학점을 구해보세요

- 90점이상 ~ 100점까지 : A
- 80점이상 90점미만 : B
- 70점이상 80점미만: C
- 60점이상 70점미만 : D
- 60점미만 : F

### □ 도전 : + 를 붙여보세요

- 95점이상 ~100점까지: A+
- 90점이상 ~95미만 : A
- 85점이상 ~90점 미만: B+
- 80점이상 ~85미만 : B
- ...
- 60점 미만 : F

# Solution

```
print("짱구의 성적표! ")
print("-"*20)

score = int(input("점수를 입력하세요 : "))
if(score>=90) : grade="A"
elif (score>=80) : grade = "B"
elif (score>=70) : grade = "C"
elif (score>=60) : grade = "D"
else : grade="F"
```

```
print("짱구의 점수는 %d이고, 학점은 %s입니다."%(score,grade))
```

#도전! 60점이상인 경우에 대해서, + 부여하기

```
if(score>=95) : grade="A+"
elif (score>=90) : grade = "A"
elif (score>=85) : grade = "B+"
elif (score>=80) : grade = "B"
#.....
```

#도 가능하지만, + , - , 0 를 모두 붙인다면 비효율적인 코드이다. 더 좋은 방법을 생각해보자!

#도전! 60점이상인 경우에 대해서, + 부여하기

```
if(score>=60):
    if(score==100 or score%10>=5) :
        grade += "+"
```

## Lab#5: 축구게임

- 사용자에게 입력을 받아 골을 막는 프로그램을 작성해 봅니다.
- 선택 : 왼쪽, 오른쪽, 중앙

어디를 수비하시겠어요?(왼쪽, 중앙, 오른쪽)중앙  
페널티 킥이 성공하였습니다.



# Solution

---

```
import random
options=["왼쪽","중앙","오른쪽"]    #options   리스트에 옵션값 저장
computer_choice = random.choice(options) # option에서 아무거나 선택
user_choice = input("어디를 수비하시겠어요?(왼쪽, 중앙, 오른쪽)")
if computer_choice == user_choice:
    print("수비에 성공하셨습니다. ")
else:
    print("페널티 킥이 성공하였습니다. ")
```

## Supp. 비교문 연습

- 두 수를 입력 받아서 두 수가 모두 짝수이면 “모두 짝수입니다.”, 두 수가 모두 홀수이면 “모두 홀수입니다.”, 그렇지 않으면 “홀수와 짝수가 섞여있습니다.”를 출력하세요
  
- 현재 월에 해당하는 정수를 입력받아 계절을 출력하는 프로그램을 작성하세요
  - 단, 1부터 12사이의 숫자가 아닌 경우에는 “잘못된 입력입니다 ” 를 출력하세요
  - 계절 예 : 봄 : 3~5, 여름 : 6~8, 가을 : 9~11, 겨울 : 12~2

### 실행 결과

첫 번째 숫자 입력 : 12  
두 번째 숫자 입력 : 24  
두 숫자가 모두 짝수입니다.

### 실행 결과

현재의 월을 입력(정수) : 3  
봄입니다.

현재의 월을 입력(정수) : 13  
잘못된 입력입니다



# 이번 장 정리

---

- `>`, `<`, `==`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.
- `Else if` → `elif`

수고하셨습니다.