

# Decisions

국민대 소프트웨어학부

# 수업목표

- if statement
- Indenting
- Logical operators: and or not
- Ternary operator

# if statement

- 만약 .. 하다면 (if),
  - If condition + statement(s)
  - Conditional statement (조건문)
    - if condition: 참(true) 또는 거짓(false) 판별
      - Boolean
  - Statement(s) : 실행문(들)

if 조건문:

    실행문

elif 조건문:

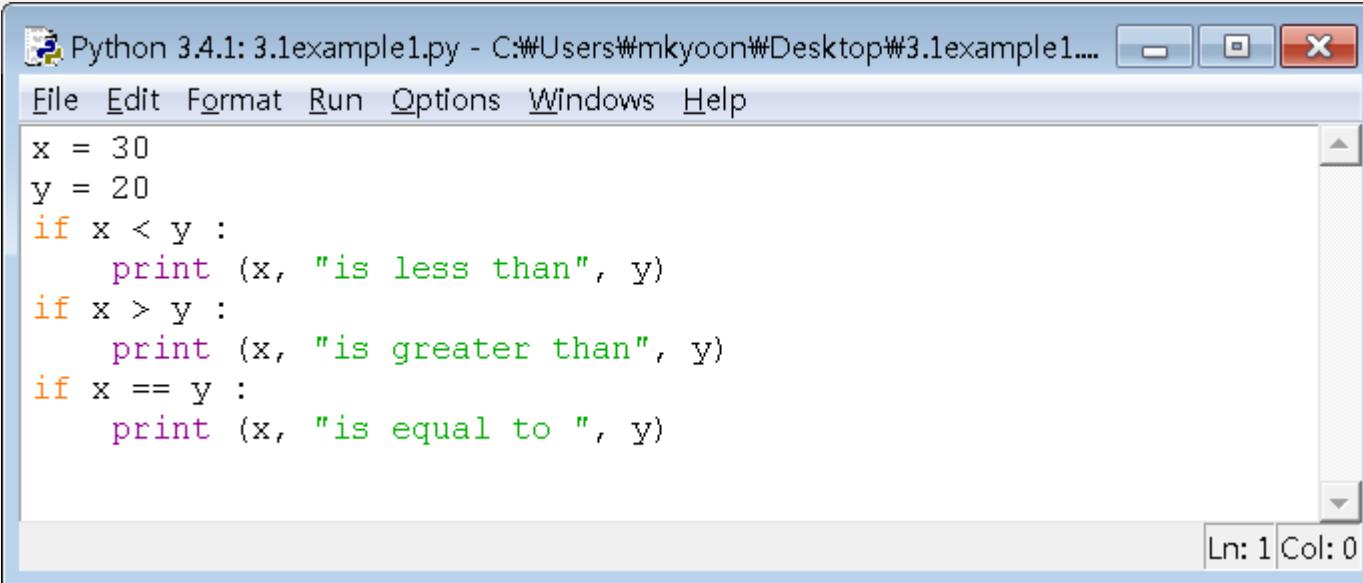
    실행문

else:

    실행문

# if statement

- 만약 .. 하다면 (if),
  - If condition + statement(s)
  - Conditional statement (조건문)
    - if condition: 참(true) 또는 거짓(false) 판별
      - Boolean
  - Statement(s) : 실행문(들)



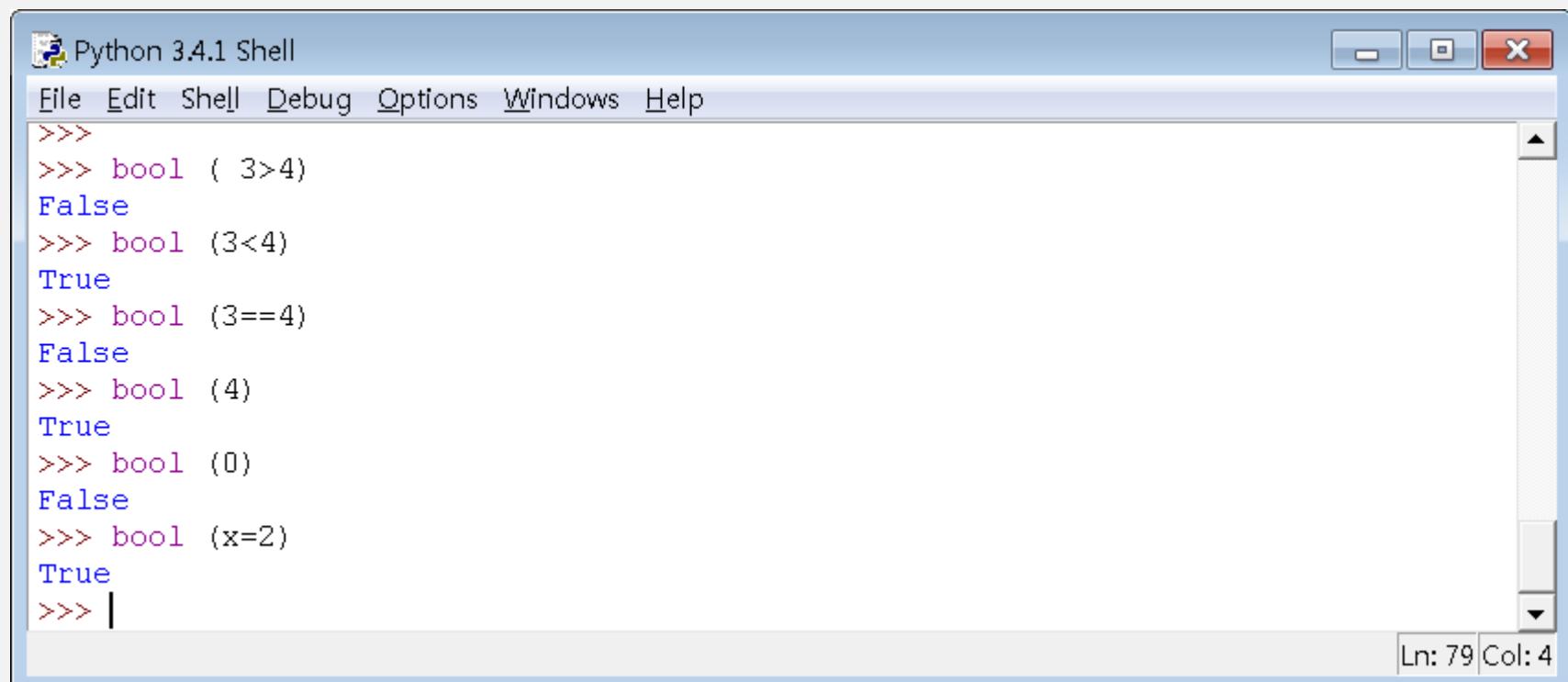
The screenshot shows a Python 3.4.1 IDE window titled "Python 3.4.1: 3.1example1.py - C:\Users\mkyoon\Desktop\3.1example1....". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code editor contains the following Python script:

```
x = 30
y = 20
if x < y :
    print (x, "is less than", y)
if x > y :
    print (x, "is greater than", y)
if x == y :
    print (x, "is equal to ", y)
```

The status bar at the bottom right indicates "Ln: 1 Col: 0".

# if statement

- 만약 .. 하다면 (if),
  - Conditional statement (조건문)
    - if condition: 참(true) 또는 거짓(false) 판별
  - '=' : 할당 연산 (assignment)
  - '==' : 비교 연산(comparison)



The screenshot shows a Windows-style application window titled "Python 3.4.1 Shell". The menu bar includes File, Edit, Shell, Debug, Options, Windows, and Help. The main window displays a series of Python shell interactions:

```
>>>
>>> bool ( 3>4)
False
>>> bool (3<4)
True
>>> bool (3==4)
False
>>> bool (4)
True
>>> bool (0)
False
>>> bool (x=2)
True
>>> |
```

In the bottom right corner, there are status indicators: Ln: 79 Col: 4.

# if statement

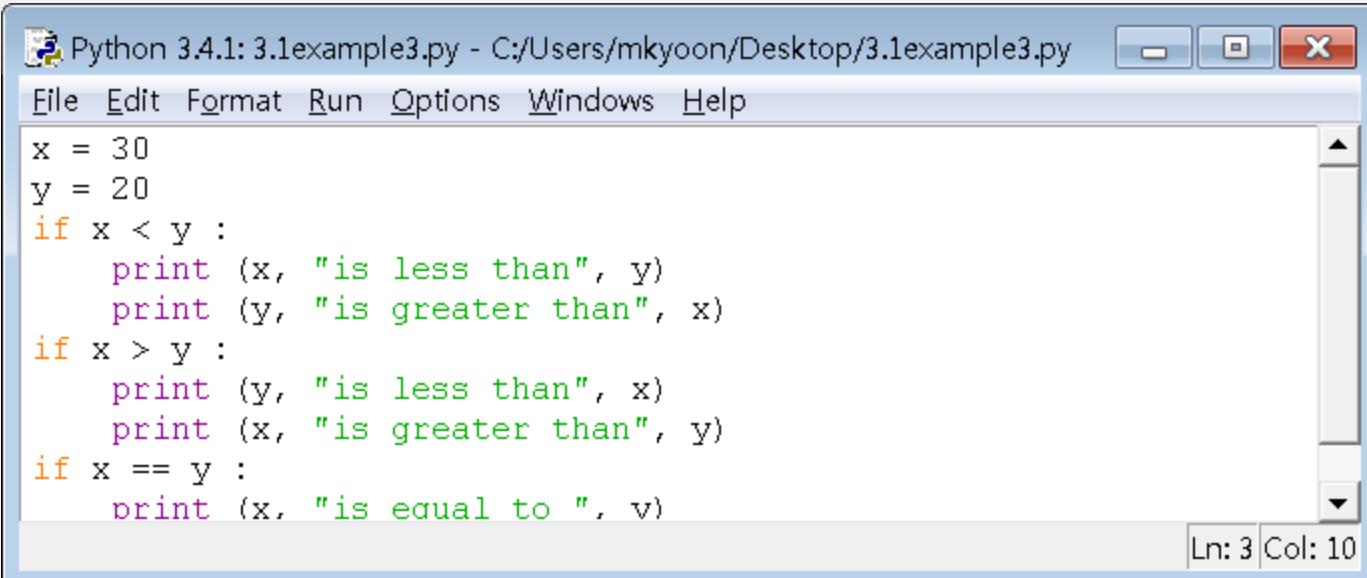
- else 문
  - 위의 조건이 모두 거짓인 경우 수행
  - 들여쓰기(indent)로 <조건이 거짓일 때 실행할 문장> 구분

```
if <조건>:  
    <조건이 참일 때 실행할 문장>  
else :  
    <조건이 거짓일 때 실행할 문장>
```

```
number = int(input("정수 입력> "))  
  
# 짝수 조건  
if number % 2 == 0 :  
    print("{} 는 짝수입니다.".format(number))  
else :  
    print("{} 는 홀수입니다.".format(number))
```

# Indenting

- Indenting (들여쓰기)
  - 파이썬 필수 요소
    - 다른 언어는 보기 좋은 용도로만 사용: C, C++, Java, ...
  - 코드블록은 들여쓰기로 결정
    - 가장 바깥쪽 블록은 1열부터 시작
    - 동일 블록은 같은 거리만큼 들여쓰기
    - 4칸씩 들여쓰기 권장



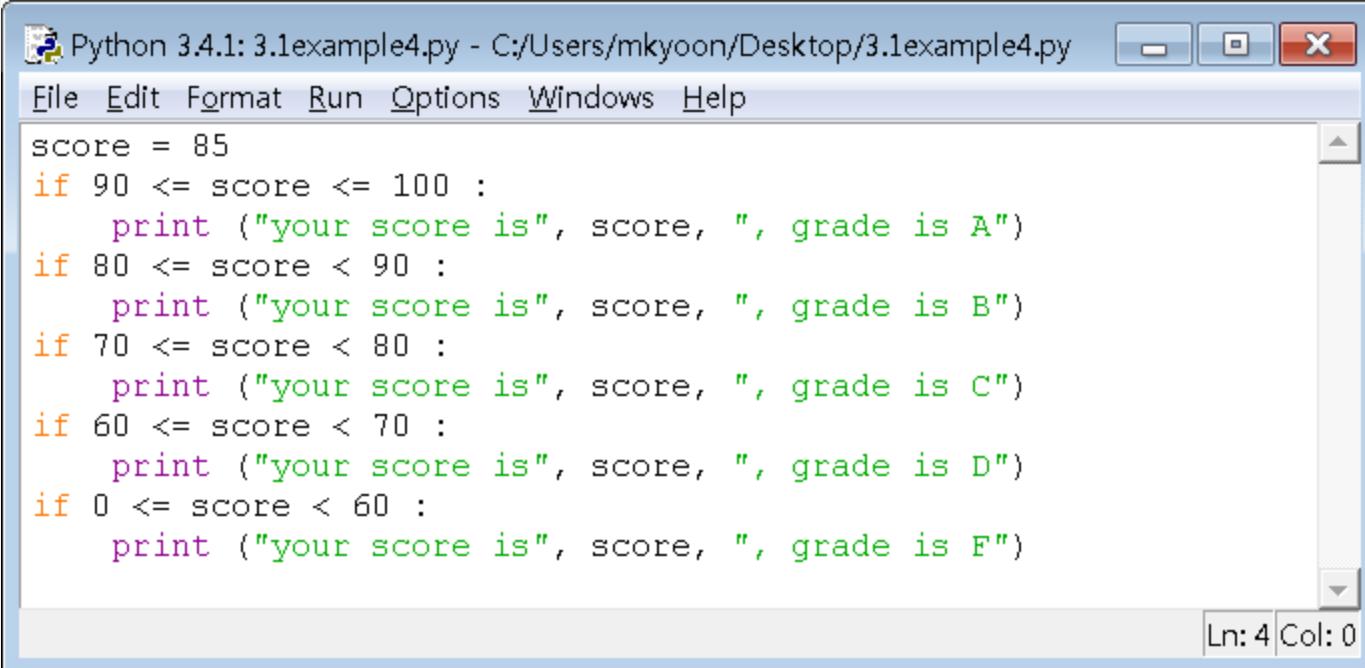
The screenshot shows a Python 3.4.1 IDE window with the title "Python 3.4.1: 3.1example3.py - C:/Users/mkyoon/Desktop/3.1example3.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code editor displays the following Python script:

```
x = 30
y = 20
if x < y :
    print (x, "is less than", y)
    print (y, "is greater than", x)
if x > y :
    print (y, "is less than", x)
    print (x, "is greater than", y)
if x == y :
    print (x, "is equal to ", v)
```

The code uses four-space indentation for blocks. The status bar at the bottom right shows "Ln: 3 Col: 10".

# Indenting

- Indenting (들여쓰기)



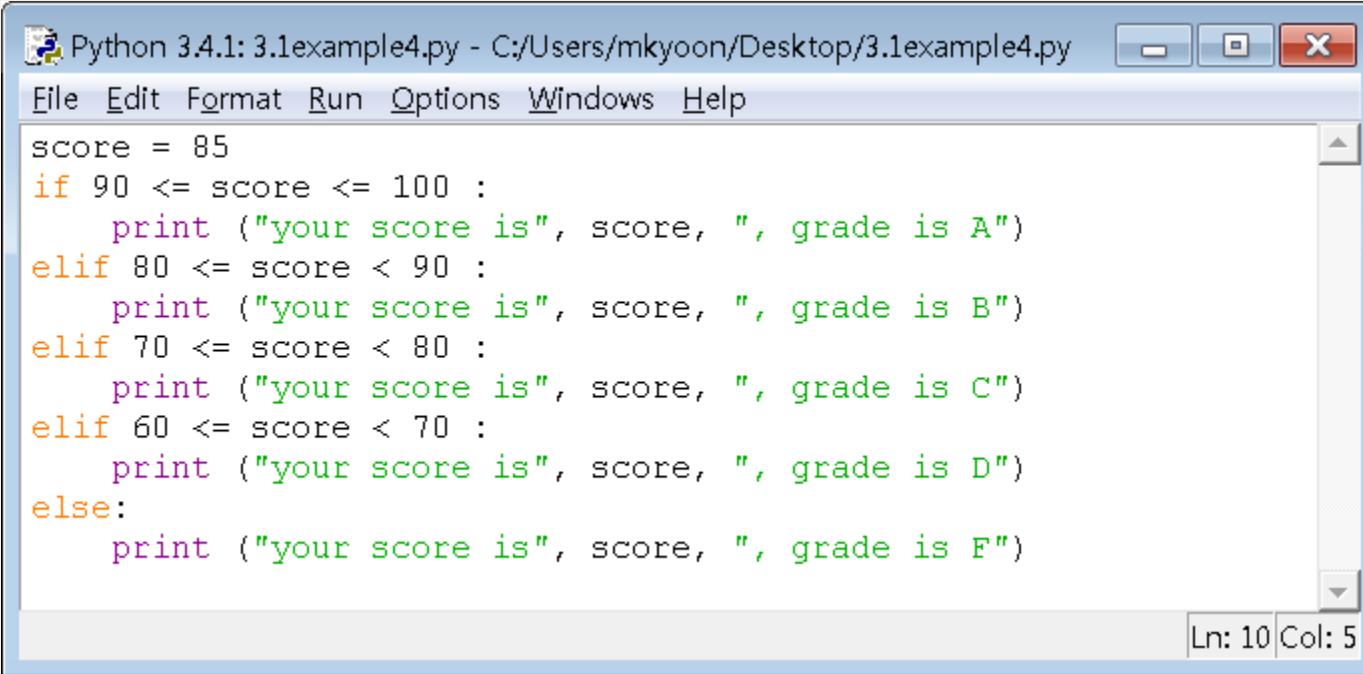
The screenshot shows a Windows-style application window titled "Python 3.4.1: 3.1example4.py - C:/Users/mkyoon/Desktop/3.1example4.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The main text area contains the following Python code:

```
score = 85
if 90 <= score <= 100 :
    print ("your score is", score, ", grade is A")
if 80 <= score < 90 :
    print ("your score is", score, ", grade is B")
if 70 <= score < 80 :
    print ("your score is", score, ", grade is C")
if 60 <= score < 70 :
    print ("your score is", score, ", grade is D")
if 0 <= score < 60 :
    print ("your score is", score, ", grade is F")
```

The status bar at the bottom right indicates "Ln: 4 Col: 0".

# Indenting

- Indenting (들여쓰기)



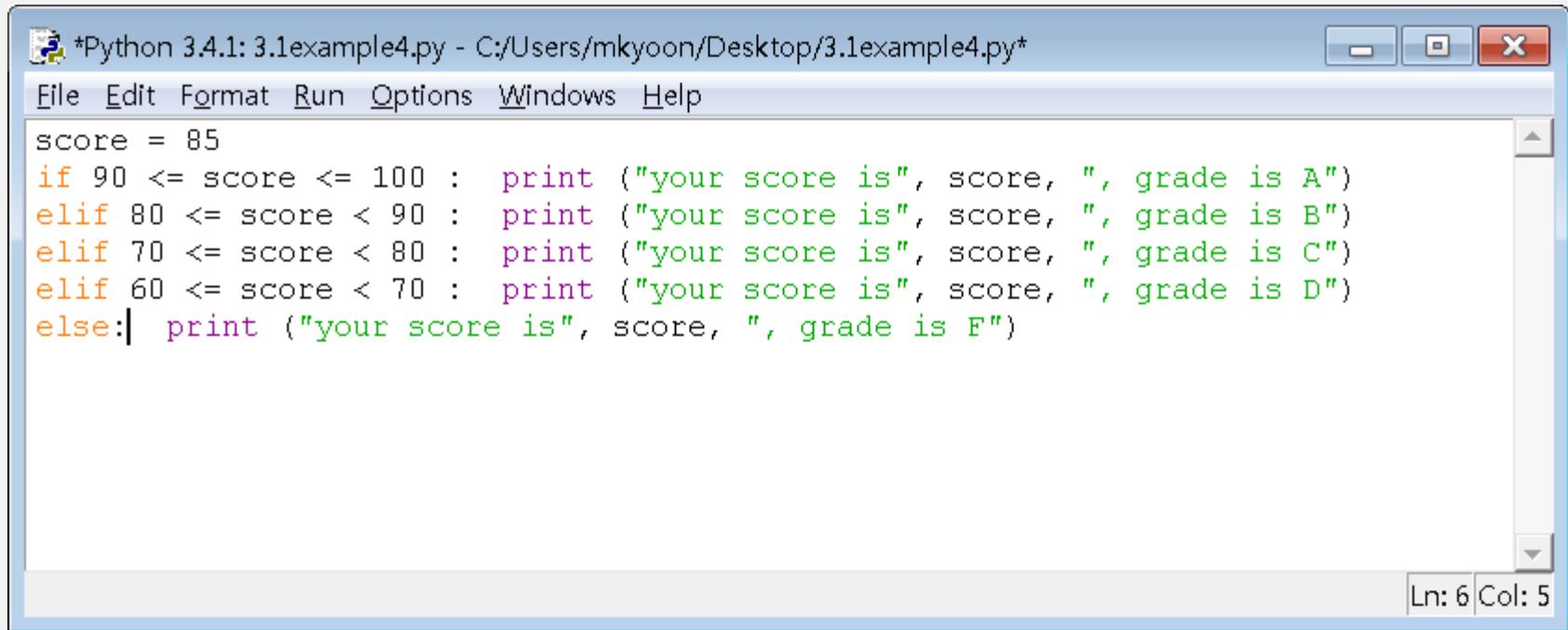
The screenshot shows a Windows-style application window titled "Python 3.4.1: 3.1example4.py - C:/Users/mkyoon/Desktop/3.1example4.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The main code area contains the following Python script:

```
score = 85
if 90 <= score <= 100 :
    print ("your score is", score, ", grade is A")
elif 80 <= score < 90 :
    print ("your score is", score, ", grade is B")
elif 70 <= score < 80 :
    print ("your score is", score, ", grade is C")
elif 60 <= score < 70 :
    print ("your score is", score, ", grade is D")
else:
    print ("your score is", score, ", grade is F")
```

The status bar at the bottom right indicates "Ln: 10 Col: 5".

# Indenting

- Indenting (들여쓰기)
  - 실행문이 하나면 한 줄 쓰기 가능



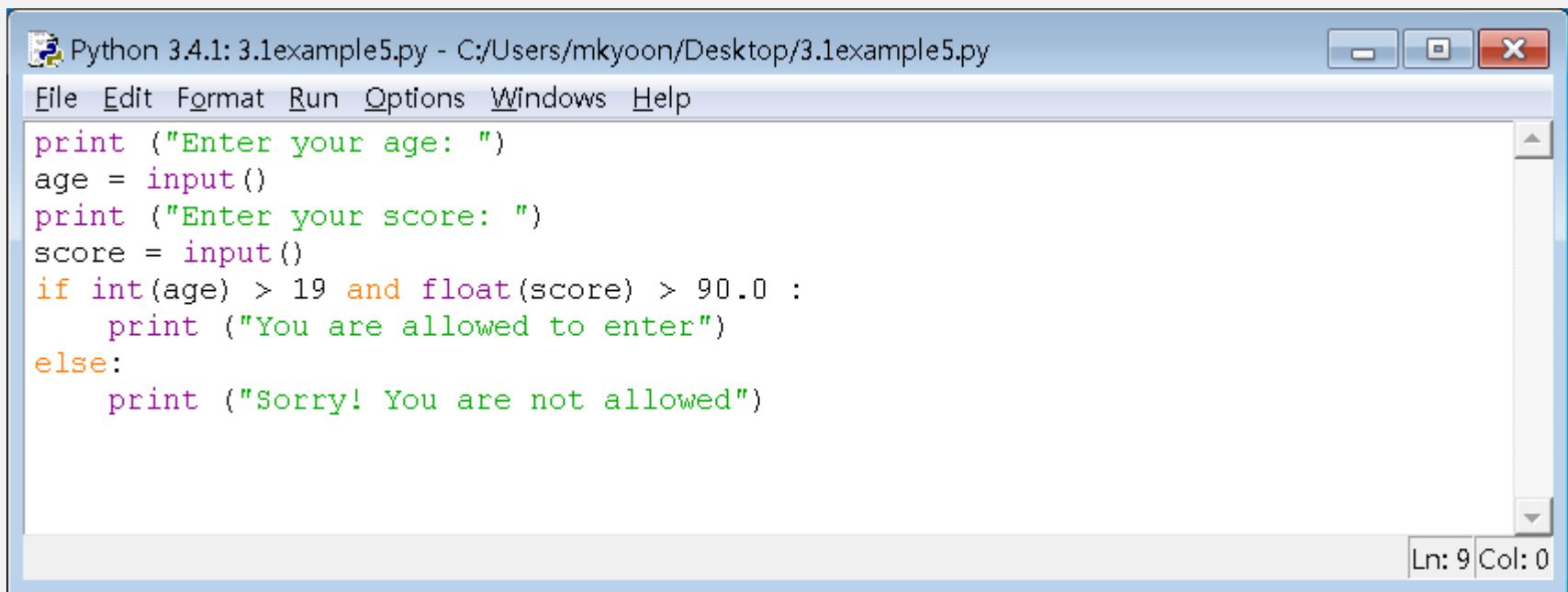
The screenshot shows a Windows Notepad window titled "\*Python 3.4.1: 3.1example4.py - C:/Users/mkyoon/Desktop/3.1example4.py\*". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code in the editor is:

```
score = 85
if 90 <= score <= 100 : print ("your score is", score, ", grade is A")
elif 80 <= score < 90 : print ("your score is", score, ", grade is B")
elif 70 <= score < 80 : print ("your score is", score, ", grade is C")
elif 60 <= score < 70 : print ("your score is", score, ", grade is D")
else:| print ("your score is", score, ", grade is F")
```

The status bar at the bottom right indicates "Ln: 6 Col: 5".

# Logical operators: and or not

- 다중 조건 테스트
  - and: 두 조건이 동시에 만족되어야 true
  - or : 두 조건 중 하나만 만족되면 true
  - not: 조건의 참/거짓 값의 반대



The screenshot shows a Python 3.4.1 IDE window titled "Python 3.4.1: 3.1example5.py - C:/Users/mkyoon/Desktop/3.1example5.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code editor contains the following Python script:

```
print ("Enter your age: ")
age = input()
print ("Enter your score: ")
score = input()
if int(age) > 19 and float(score) > 90.0 :
    print ("You are allowed to enter")
else:
    print ("Sorry! You are not allowed")
```

The status bar at the bottom right indicates "Ln: 9 Col: 0".

# Logical operators: and or not

- 다중 조건 테스트
  - 논리 연산자 (Boolean)

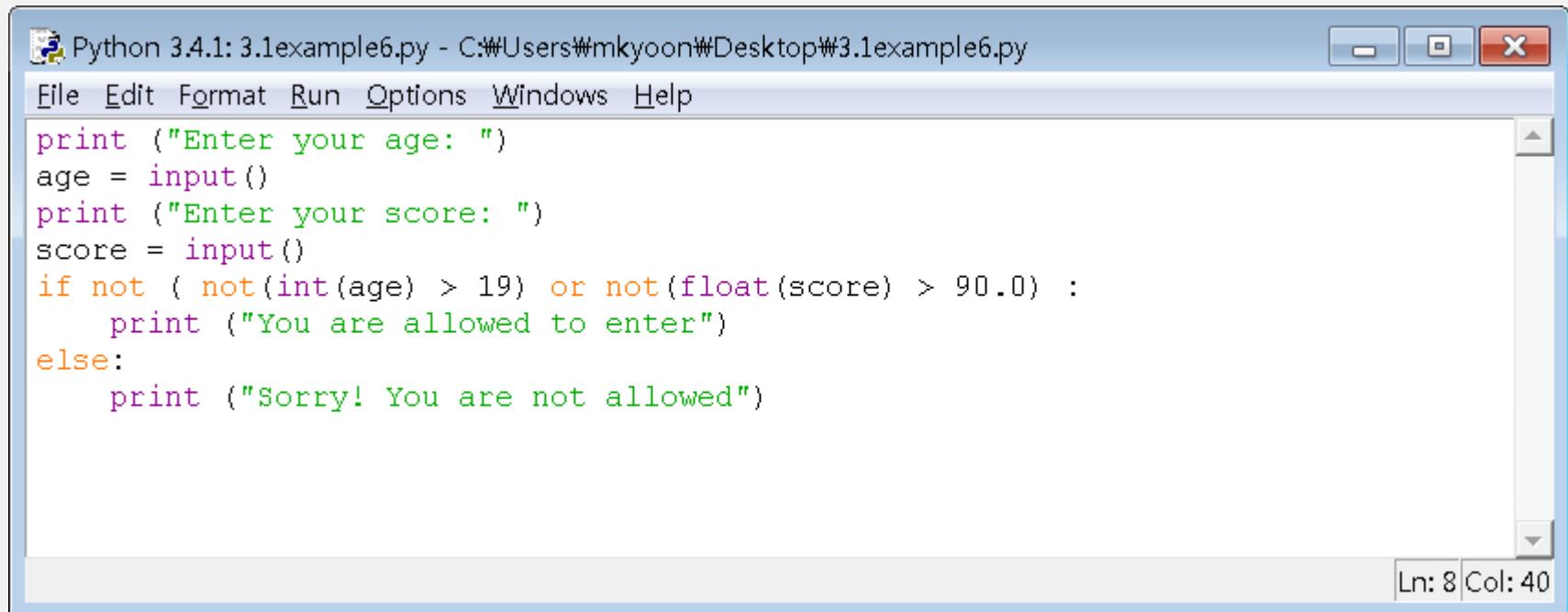
```
x = 10  
is_under_20 = x < 20  
print("is_under_20", is_under_20)  
print("not is_under_20", not is_under_20)
```

```
print(False and False)  
print(False and True)  
print(True and False)  
print(True and True)  
print()
```

```
print(False or False)  
print(False or True)  
print(True or False)  
print(True or True)
```

# Logical operators: and or not

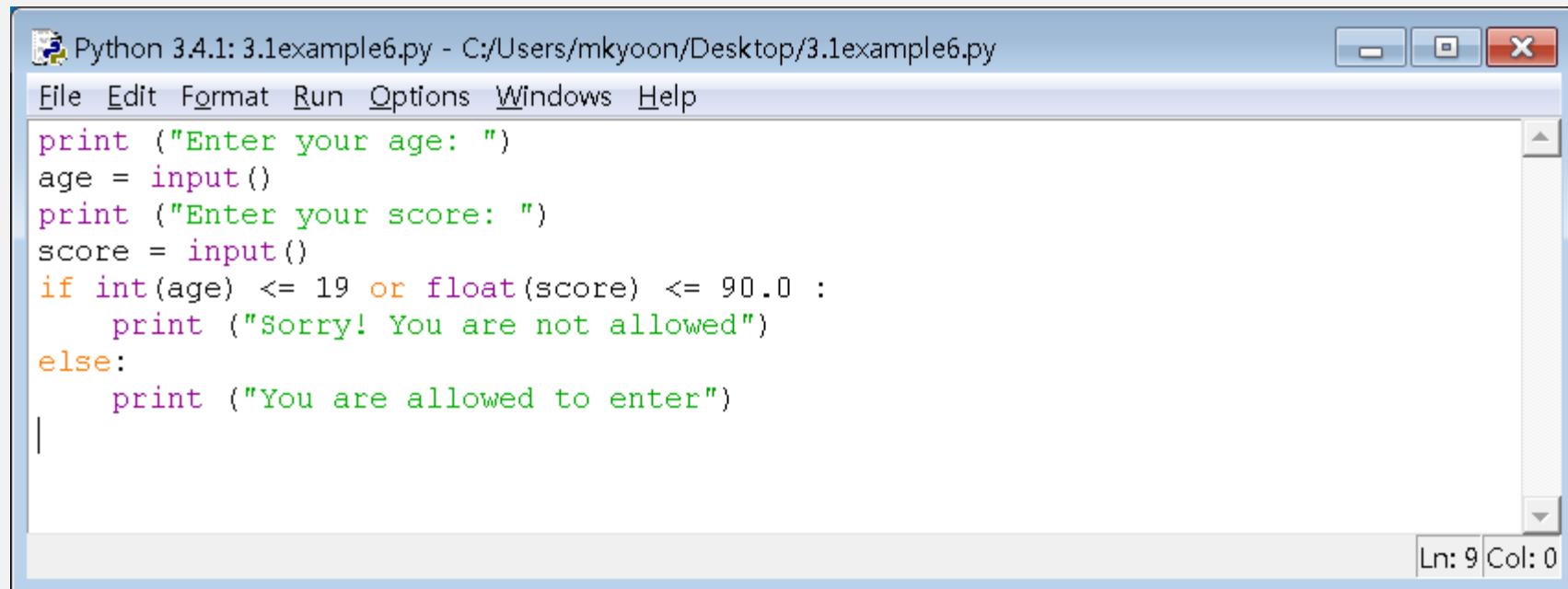
- 다중 조건 테스트
  - and: 두 조건이 동시에 만족되어야 true
  - or : 두 조건 중 하나만 만족되면 true
  - not: 조건의 참/거짓 반대
  - A and B = not ( (not A) or (not B))



```
Python 3.4.1: 3.1example6.py - C:\Users\mkyoon\Desktop\3.1example6.py
File Edit Format Run Options Windows Help
print ("Enter your age: ")
age = input()
print ("Enter your score: ")
score = input()
if not ( not(int(age) > 19) or not(float(score) > 90.0) :
    print ("You are allowed to enter")
else:
    print ("Sorry! You are not allowed")
Ln: 8 Col: 40
```

# Logical operators: and or not

- 다중 조건 테스트
  - and: 두 조건이 동시에 만족되어야 true
  - or : 두 조건 중 하나만 만족되면 true
  - not: 조건의 참/거짓 반대
  - $A \text{ and } B = \text{not} (\text{not } A \text{ or } \text{not } B)$



The screenshot shows a Windows-style terminal window titled "Python 3.4.1: 3.1example6.py - C:/Users/mkyoon/Desktop/3.1example6.py". The window contains the following Python code:

```
print ("Enter your age: ")
age = input()
print ("Enter your score: ")
score = input()
if int(age) <= 19 or float(score) <= 90.0 :
    print ("Sorry! You are not allowed")
else:
    print ("You are allowed to enter")
```

The status bar at the bottom right indicates "Ln: 9 Col: 0".

# Ternary operator

- Ternary operator (삼항 연산자)
  - if 문을 대신할 수 있는 연산자

```
*Python 3.4.1: 3.1example8.py - C:/Users/mkyoon/Desktop/3.1example8.py*
File Edit Format Run Options Windows Help
x = 20
if x > 10:
    y = x*2
else:
    y = x/2
print (y)

Ln: 5 Col: 4
```

```
Python 3.4.1: 3.1example8.py - C:/Users/mkyoon/Desktop/3.1example8.py
File Edit Format Run Options Windows Help
x = 20
y = x*2 if x > 10 else x/2
print (y)

Ln: 4 Col: 0
```

# 실습

- 실습

- datetime 을 이용하여, 현재 시각이 오전인지 오후인지 구하는 코드를 작성하세요.

```
import datetime

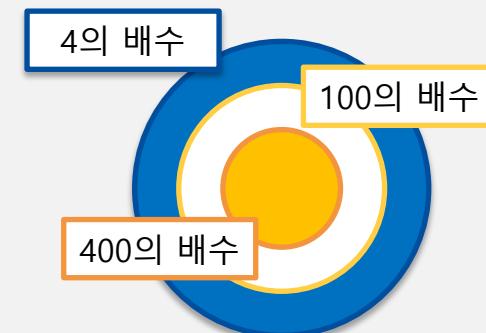
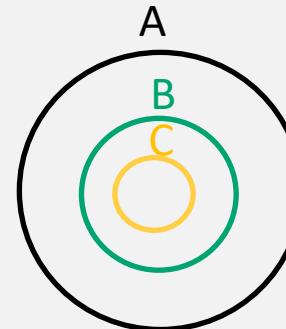
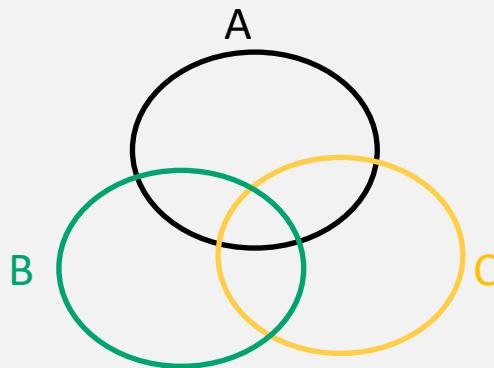
# 현재 날짜/시간을 구합니다.
now = datetime.datetime.now()

# 오전 구분
if now.hour < 12 :
    print("현재 시간은 {HOUR}시로 오전입니다.".format(HOUR = now.hour))
else :
    print("현재 시간은 {HOUR}시로 오후입니다.".format(HOUR=now.hour))
```

<https://docs.python.org/3/library/datetime.html>

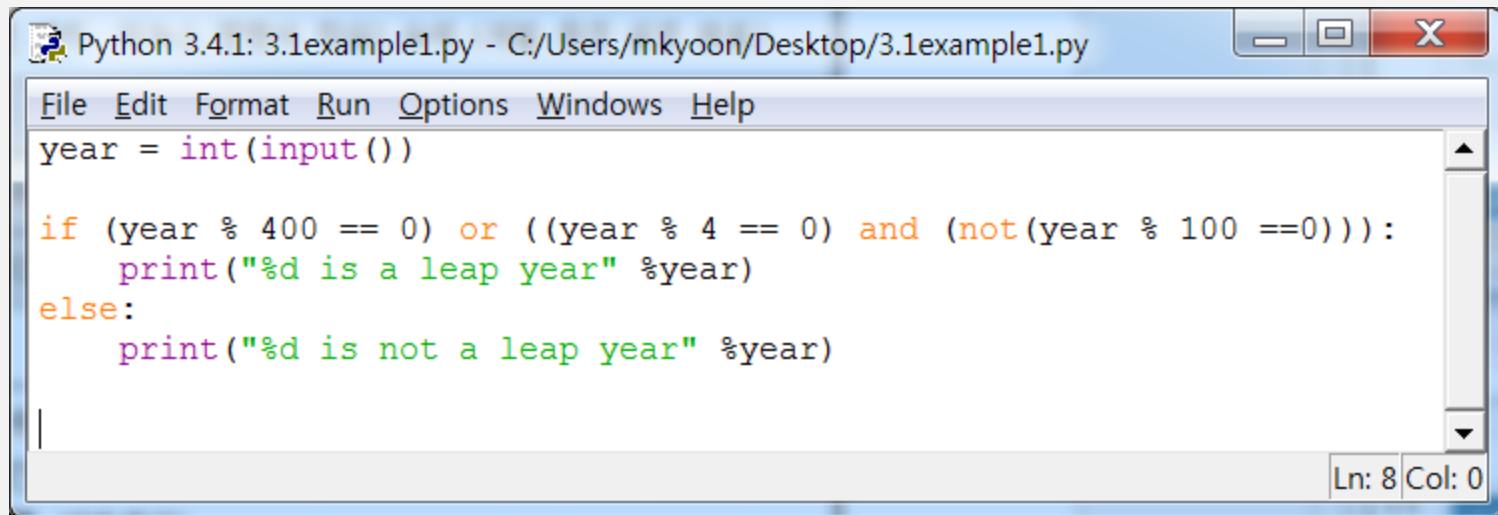
# 실습

- 윤년 판별
  - 입력되는 연도가 윤년인지 판별하는 코드를 작성하시오.
    - 4로 나누어지는 해는 윤년이다.
    - 그러나, 100으로 나누어지는 해는 윤년이 아니다.
    - 그러나, 400으로 나누어지는 해는 다시 윤년이다.
  - 논리적 생각 → 집합, 뱀다이어그램
    - A: 4로 나누어지는 연도의 집합
    - B: 100으로 나누어지는 연도의 집합
    - C: 400으로 나누어지는 연도의 집합



# 실습

- 윤년 판별



The screenshot shows a window titled "Python 3.4.1: 3.1example1.py - C:/Users/mkyoon/Desktop/3.1example1.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code in the editor is:

```
year = int(input())

if (year % 400 == 0) or ((year % 4 == 0) and (not(year % 100 ==0 ))):
    print("%d is a leap year" %year)
else:
    print("%d is not a leap year" %year)
```

The status bar at the bottom right indicates "Ln: 8 Col: 0".

# 실습

- 삼각형 판별
  - 삼각형의 세변의 길이( $a \leq b \leq c$ )를 입력 받은 다음, 입력 받은 삼각형이 예각 삼각형인지, 둔각 삼각형인지, 직각 삼각형인지 판별하는 파이썬 코드를 작성하세요. ( 단 삼각형을 만들 수 없는 경우는 없다고 가정 )

# 실습

- 

## 삼각형 판별

- 삼각형의 세변의 길이( $a \leq b \leq c$ )를 입력 받은 다음, 입력 받은 삼각형이 예각 삼각형인지, 둔각 삼각형인지, 직각 삼각형인지 판별하는 파이썬 코드를 작성하세요. ( 단 삼각형을 만들 수 없는 경우는 없다고 가정 )

```
# 삼각형 판단 하기
```

```
a = int(input("a> "))
b = int(input("b> "))
c = int(input("c> "))

if a ** 2 + b ** 2 == c ** 2 :
    print("직각삼각형 입니다.")
elif a ** 2 + b ** 2 > c ** 2 :
    print("둔각삼각형 입니다.")
else :
    print("예각삼각형 입니다.")
```

# 실습

- `datetime` 을 이용하여, 오늘의 계절을 구분하는 코드를 작성 하세요
  - 봄 : 3월, 4월, 5월
  - 여름 : 6월, 7월, 8월
  - 가을 : 9월, 10월, 11월
  - 겨울 : 12월, 1월, 2월

# 실습

- datetime 을 이용하여, 오늘의 계절을 구분하는 코드를 작성 하세요
  - 봄 : 3월, 4월, 5월
  - 여름 : 6월, 7월, 8월
  - 가을 : 9월, 10월, 11월
  - 겨울 : 12월, 1월, 2월

```
import datetime

# 현재 날짜/시간을 구합니다.
now = datetime.datetime.now()

# 봄
if 3 <= now.month <= 5 :
    print("이번달은 {MONTH}월로 봄입니다.".format(MONTH=now.month))
# 여름
elif 6 <= now.month <= 8 :
    print("이번달은 {MONTH}월로 여름입니다.".format(MONTH=now.month))
# 가을
elif 9 <= now.month <= 11 :
    print("이번달은 {MONTH}월로 가을입니다.".format(MONTH=now.month))
# 겨울
#elif now.month == 12 or 1 <= now.month <= 2 :
else :
    print("이번달은 {MONTH}월로 겨울입니다.".format(MONTH=now.month))
```