

Decisions

By 박수현

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

3.1example1.py

File Edit Format Run Options Window Help

```
# x, y 두개의 값을 input()함수를 이용하여 받아들이м
x = float(input(">> x = "))
y = float(input(">> y = "))

print("\n>> 두 개의 수 x = {}, y = {} 크기 비교\n".format(x, y))

print("# if 만 이용")
if x < y :
    print("> x : {} is less than y : {}".format(x, y))

if x > y :
    print("> x : {} is greater than y : {}".format(x, y))

if x == y :
    print("> x : {} is equal to y : {}".format(x, y))

print("\n## nested if 이용")
if x < y :
    print("> x : {} is less than y : {}".format(x, y))

elif x > y : #nested if
    print("> x : {} is greater than y : {}".format(x, y))

elif x == y :
    print("> x : {} is equal to y : {}".format(x, y))
```

```
>> x = 10
>> y = 20
```

>> 두 개의 수 x = 10.0, y = 20.0 크기 비교

```
# if 만 이용
> x : 10.0 is less than y : 20.0
```

```
## nested if 이용
> x : 10.0 is less than y : 20.0
```

>>>

```
===== RESTART: C:\W소사\강의예제\W
=====
```

```
>> x = 10
>> y = 5
```

>> 두 개의 수 x = 10.0, y = 5.0 크기 비교

```
# if 만 이용
> x : 10.0 is greater than y : 5.0
```

```
## nested if 이용
> x : 10.0 is greater than y : 5.0
```

>>>

```
===== RESTART: C:\W소사\강의예제\W
=====
```

```
>> x = 10
>> y = 10
```

>> 두 개의 수 x = 10.0, y = 10.0 크기 비교

```
# if 만 이용
> x : 10.0 is equal to y : 10.0
```

```
## nested if 이용
> x : 10.0 is equal to y : 10.0
```

if statement

- 만약 .. 하다면 (if),
 - ‘=’ : 할당 연산 (assignment)
 - ‘==’: 비교 연산 (comparison)

Nested if statement

- 중첩(nesting)된 if-else문
 - if-else 문장 안에 다른 if-else 문장을 넣을 수 있음

전체적인 구조



```
if 조건1 :
```

```
    문장_A
```

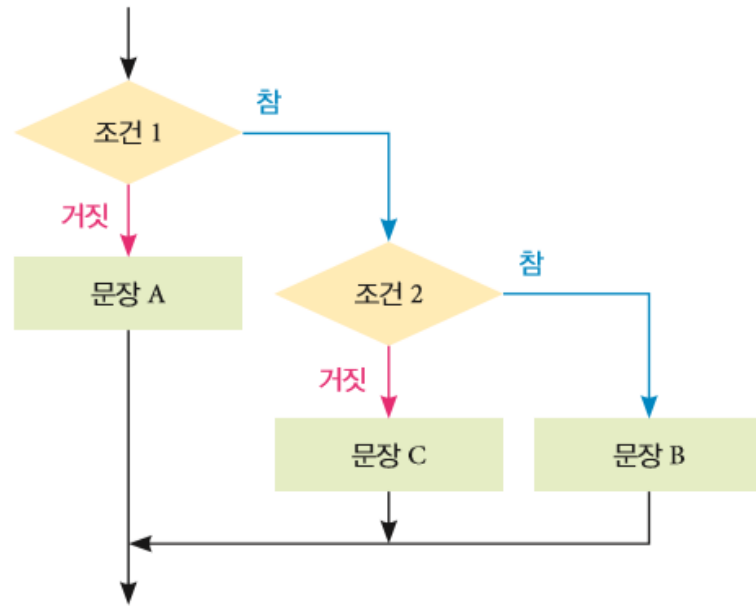
```
else:
```

```
    if 조건2 :
```

```
        문장_B
```

```
    else:
```

```
        문장_C
```



Nested if statement

```
apple.py
File Edit Format Run Options Window Help
appleQuality = input("> 사과의 상태를 다음의 값으로 입력하십시오. 신선(1), 시들(2) ")
if appleQuality == '1':
    print("\n>> 사과가 신선하군요 !! ")
    applePrice = int(input("> 사과 1개에 얼마인가요 ? : "))
    print("> 사과 1개의 가격 : {}원".format(applePrice))
    if applePrice < 1000:
        print("\n>> 신선한 사과의 가격이 저렴하네요.")
    else:
        print("\n>> 사과는 신선한대 가격이 비싸네요.")
    num_of_buying = int(input(">> 사고자 하는 사과 갯수를 입력하세요 : "))
    print(">> {}개 주세요".format(num_of_buying))
    total_price = applePrice * num_of_buying
    print(" 총 가격은 ", total_price, "원 입니다.")
else:
    print("\n>> 사과가 신선하지 않네요. 사지 않을래요")
```

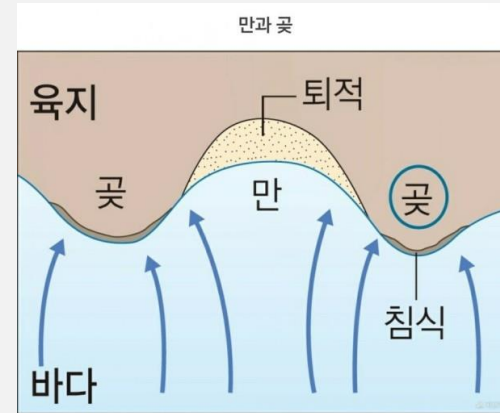
```
> 사과의 상태를 다음의 값으로 입력하십시오. 신선(1), 시들(2) : 1
>> 사과가 신선하군요 !!
> 사과 1개에 얼마인가요 ? : 3000
> 사과 1개의 가격 : 3000원

>> 사과는 신선한대 가격이 비싸네요.
>> 사고자 하는 사과 갯수를 입력하세요 : 10
>> 10개 주세요
    총 가격은 30000 원 입니다.

===== RESTART: C:\W소사W강의예제Wapple.py =====
> 사과의 상태를 다음의 값으로 입력하십시오. 신선(1), 시들(2) : 2
>> 사과가 신선하지 않네요. 사지 않을래요
```

Indenting

- Indenting (들여쓰기)
 - Python 필수 요소
 - 다른 언어는 보기 좋은 용도로만 사용: C, C++, Java, ...
 - **Code block**은 들여쓰기로 결정
 - 가장 바깥쪽 block은 1열부터 시작
 - 동일 block은 같은 거리만큼 들여쓰기
 - 4칸(column)씩 들여쓰기 권장



https://search.naver.com/search.naver?sm=tab_hy.top&where=image&query=%ED%95%B4%EC%95%88%EC%84%A0+%EB%A7%8C+&oquery=%ED%95%B4%EC%95%88%EC%84%A0+%EB%A7%8C+%EC%A7%80%EC%A7%88%ED%95%99+%EA%B7%B8%EB%A6%BC&ti=U1JCjSp0YihssuXy8VNssss%2BR-308934#imgId=blog4200382%7C71%7C221976434429_638047799&vType=rollout

```
3.1example3.py ·
File Edit Format Run Optic
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)
```

```
1 public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
2 {
3     if (parameter is string)
4     {
5         var dict = value as Dictionary<string, object>;
6         if (dict != null)
7         {
8             string[] parameters = ((string)parameter).Split(':');
9             string key = parameters[0];
10            if (parameters.Length >= 2)
11            {
12                string format = parameters[1];
13                return String.Format("{0:" + format + "}", dict[key]);
14            }
15            return dict[key];
16        }
17    }
18    return string.Empty;
19 }
```


Indenting

- Indenting (들여쓰기)

3.1example4-0.py

File Edit Format Run Options Window Help

```
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")
```

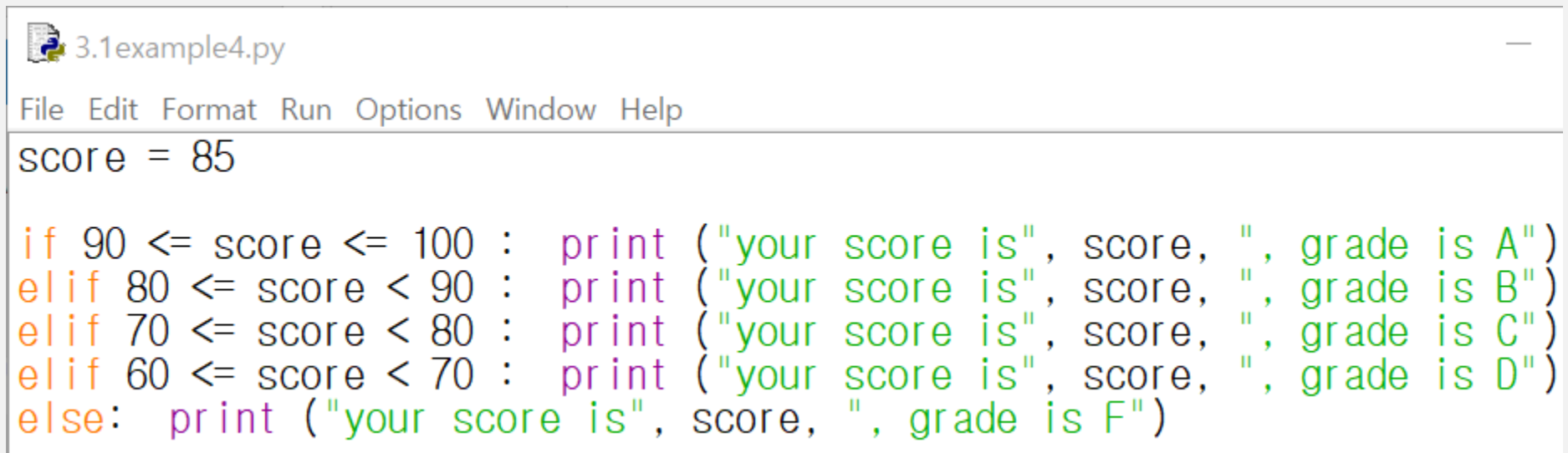
```
if 60 <= score < 70 :  
    print ("your score is", score, ", grade is D")
```

```
if 0 <= score < 60 :  
    print ("your score is", score, ", grade is F")
```

```
your score is 85 , grade is B  
>>>
```

Indenting

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



```
3.1example4.py
File Edit Format Run Options Window Help
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")
```

```
your score is 85 , grade is B
>>>
```

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
- A and B = not ((not A) or (not B))

```
= RESTART: C:\W소사W강의예제W3.1example5.py
> 나이 : 26
> 점수 : 96

>> 입력된 나이 = 26
>> 입력 점수 = 96

>> 축하합니다 ! 합격하셨습니다.
>>

===== RESTART: C:\W소사W강의예
=====
> 나이 : 29
> 점수 : 86

>> 입력된 나이 = 29
>> 입력 점수 = 86

>> 예비합격자 입니다.
>>

===== RESTART: C:\W소사W강의예
=====
> 나이 : 45
> 점수 : 99

>> 입력된 나이 = 45
>> 입력 점수 = 99

>> 미안합니다. 합격자 명단에 없습니다.
```

```
3.1example5.py
File Edit Format Run Options Window Help

# 나이가 19세 이상 40 미만이고 90점 이상이면 합격
# 나이가 19세 이상 40 미만이고 85점 이상이면 예비 합격자
# 나머지 조건은 불합격

age = input("> 나이 : ")
score = input("> 점수 : ")

print("\n>> 입력된 나이 = ", age)
print(">> 입력 점수 = ", score)

if (19 <= int(age) < 40) and float(score) >= 90.0 :
    print ("\n>> 축하합니다 ! 합격하셨습니다.")

elif (19 <= int(age) < 40) and (85 <= float(score) < 90.0):
    print ("\n>> 예비합격자 입니다.")

else:
    print ("\n>> 미안합니다. 합격자 명단에 없습니다.")
```

Ternary operator

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

```
3.1example8.py
File Edit Format Run Options Window
1 x = int(input(">> x = "))
2
3 # binary operator
4 if x > 10 :
5     y = x * 2
6 else :
7     y = x / 2
8
9 print("1) y = ", y)
10
11 # ternary operator
12 y = x*2 if x > 10 else x/2
13
14 print("2) y = ", y)
15
```

```
>> x = 20
1) y = 40
2) y = 40
>>>
=====
>> x = 8
1) y = 4.0
2) y = 4.0
```

실습

- 실습

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

```
Morning-Afternoon.py
File Edit Format Run Options Window Help
1 import datetime
2
3 # 현재 날짜/시간을 구합니다.
4 now = datetime.datetime.now()
5
6 print(">> 컴퓨팅 디바이스의 현재시간을 읽어옵니다.")
7 print(">> 현재시간 : ", now)
8
9 # 오전 / 오후 구분
10 if now.hour < 12 :
11     print(">> 현재 시간은 {HOUR}시로 오전입니다.".format(HOUR = now.hour))
12
13 else :
14     print(">> 현재 시간은 {HOUR}시로 오후입니다.".format(HOUR=now.hour))
15
16 print("\nEND\n")
17
```

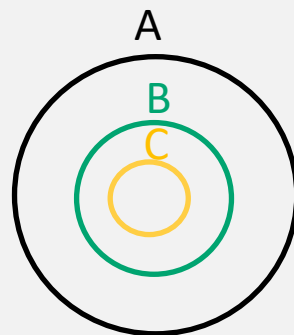
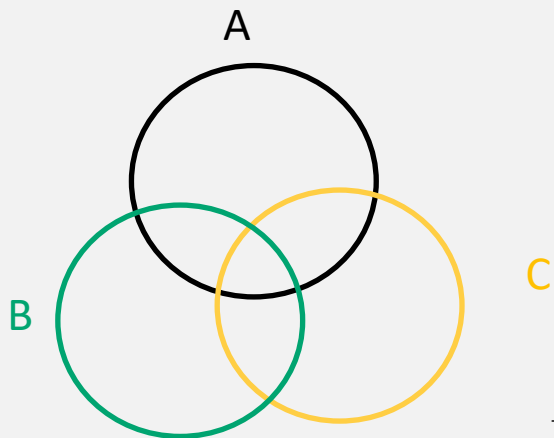
```
>> 컴퓨팅 디바이스의 현재시간을 읽어옵니다.
>> 현재시간 : 2023-09-24 22:51:55.400313
>> 현재 시간은 22시로 오후입니다.

END
```

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

실습

- 윤년(leap year) 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, 밴다이어그램
 - A: 4로 나누어지는 연도의 집합
 - B: 100으로 나누어지지 않는 연도의 집합
 - C: 400으로 나누어지는 연도의 집합



실습

- 윤년 판별

leap_year.py

File Edit Format Run Options Window Help

```
1 print()
2 year = int(input("> 연도를 입력하세요 >> "))
3
4 if (year % 400 == 0) or ((year % 4 == 0) and (not(year % 100 == 0))) :
5     print("\n>> %d is a leap year" %year)
6
7 else :
8     print("\n>>%d is not a leap year" %year)
9
```

A: 4로 나누어지는 연도의 집합

B: 100으로 나누어지지 않는 연도의 집합

C: 400으로 나누어지는 연도의 집합

> 연도를 입력하세요 >> 2018

>>2018 is not a leap year

>>>

===== f
=====

> 연도를 입력하세요 >> 2020

>> 2020 is a leap year

>>>

===== f
=====

> 연도를 입력하세요 >> 2200

>>2200 is not a leap year

>>>

실습

- Id 검사

check_id.py

File Edit Format Run Options Window Help

```
1 user_list = ['김국민', '박소프', '이소용', '나신입']
2 password = '$12%45'
3
4 name = input('>> ID : ')
5
6 if name in user_list:
7     pw = input('>> PW : ')
8
9     if pw == '$12%45':
10         print("\n>> {}님 환영합니다.".format(name))
11     else:
12         print("\n>> 잘못된 pw입니다 ")
13 else:
14     print("\n>> {}님은 승인되지 않은 사용자입니다.".format(name))
15
```

```
=====
>> ID : 박소프
>> PW : $12%45

>> 박소프님 환영합니다.
>>>
>>>
===== RESTART: C:/
=====
>> ID : 김공대

>> 김공대님은 승인되지 않은 사용자입니다.
>>>
===== RESTART: C:/
=====
>> ID : 김국민
>> PW : $$$$

>> 잘못된 pw입니다
^
```


실습

- 2차 방정식 근을 계산

- 사용자에게 2차 방정식의 계수 a,b,c 를 입력하도록 함
- 만약 a가 0이면 근은 -c/b
- 만약 판별식 ($b^2 - 4ac$)가 음수이면 실근이 존재하지 않음
- 위의 조건을 만족하지 않으면 근의 공식을 적용

```
>> 삼각형의 세변의 값 A, B, C를 입력하시오.  
A = 4  
B = 7  
C = 8  
  
>> A = 4.0, B = 7.0, C = 8.0  
>> 판별식 D 값 = -79.0  
  
>> 실근이 존재하지 않음
```

```
>> 삼각형의 세변의 값 A, B, C를 입력하시오.  
A = 1  
B = 2  
C = 1  
  
>> A = 1.0, B = 2.0, C = 1.0  
>> 판별식 D 값 = 0.0  
  
>> x = -1.0
```

```
>> 2차 방정식의 계수 A, B, C를 입력하시오.
```

```
A = 1  
B = 3  
C = 1
```

```
>> A = 1.0, B = 3.0, C = 1.0  
>> 판별식 D 값 = 5.0
```

```
>> x1 = -0.3819660112501051  
>> x2 = -2.618033988749895
```

quad_eq.py

File Edit Format Run Options Window Help

```
import math
```

```
print(">> 2차 방정식의 계수 A, B, C를 입력하시오.")
```

```
A = float(input("A = "))
```

```
B = float(input("B = "))
```

```
C = float(input("C = "))
```

```
print("\n>> A = {}, B = {}, C = {}".format(float(A), float(B), float(C)))
```

```
D = B * B - 4 * A * C
```

```
print(">> 판별식 D 값 = ", D)
```

```
if A == 0:
```

```
    print("\n>> x=", -C/B)
```

```
if D == 0:
```

```
    print("\n>> x =", -B / (2.0 * A))
```

```
elif D > 0:
```

```
    print("\n>> x1 =", (-B + math.sqrt(D)) / (2.0 * A))
```

```
    print(">> x2 =", (-B - math.sqrt(D)) / (2.0 * A))
```

```
else:
```

```
    print("\n>> 실근이 존재하지 않음")
```

실습

- 삼각형 판별
 - 삼각형의 세변의 길이($a \leq b \leq c$)를 입력 받은 다음, 입력 받은 삼각형이 예각 삼각형(an acute(angled) triangle)인지, 둔각 삼각형(an obtuse triangle)인지, 직각 삼각형(a right-angled triangle)인지를

판별하는 python code를 작성하세요.

실습

check_triangle_type.py

File Edit Format Run Options Window Help

```
1 # 삼각형 판단하기
2
3 print(">> 삼각형의 세변의 값 a, b, c를 입력하세요")
4 print("조건 : a <= b <= c")
5
6 a = float(input(">> a : "))
7 b = float(input("b : "))
8 c = float(input("c : "))
9
10 print("\n>> a = {}, b = {}, c = {}".format(a,b,c))
11
12 if a <= b <= c :
13     if a ** 2 + b ** 2 == c ** 2 :
14         print("\n>> 직각삼각형 입니다.")
15     elif a ** 2 + b ** 2 > c ** 2 :
16         print("\n>> 둔각삼각형 입니다.")
17     else :
18         print("\n>> 예각삼각형 입니다.")
19 else:
20     print("\n$ a <= b <= c 의 조건을 만족해야 합니다")
21
22
23
24
```

```
>> 삼각형의 세변의 값 a, b, c를 입력하세요
조건 : a <= b <= c
>> a : 1
b : 2
c : 3
```

```
>> a = 1.0, b = 2.0, c = 3.0
```

```
>> 예각삼각형 입니다.
```

```
>>>
```

```
===== RESTART: C:/과소사/
=====
```

```
>> 삼각형의 세변의 값 a, b, c를 입력하세요
조건 : a <= b <= c
```

```
>> a : 3
b : 4
c : 5
```

```
>> a = 3.0, b = 4.0, c = 5.0
```

```
>> 직각삼각형 입니다.
```

```
>>>
```

```
===== RESTART: C:/과소사/
=====
```

```
>> 삼각형의 세변의 값 a, b, c를 입력하세요
조건 : a <= b <= c
```

```
>> a : 1
b : 3
c : 2
```

```
>> a = 1.0, b = 3.0, c = 2.0
```

```
$ a <= b <= c 의 조건을 만족해야 합니다
>>>
```

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

season_decision.py

File Edit Format Run Options Window Help

```
1 import datetime
2
3 # 현재 날짜/시간을 구합니다.
4 now = datetime.datetime.now()
5 print("\n>> now = {}".format(now))
6
7 # 봄
8 if 3 <= now.month <= 5 :
9     print(">> 이번 달은 {MONTH}월로 봄입니다.".format(MONTH=now.month))
10
11 # 여름
12 elif 6 <= now.month <= 8 :
13     print(">> 이번 달은 {MONTH}월로 여름입니다.".format(MONTH=now.month))
14
15 # 가을
16 elif 9 <= now.month <= 11 :
17     print(">> 이번 달은 {MONTH}월로 가을입니다.".format(MONTH=now.month))
18
19 # 겨울
20 elif now.month == 12 or 1 <= now.month <= 2 :
21     print(">> 이번 달은 {MONTH}월로 겨울입니다.".format(MONTH=now.month))
22
```

```
>> now = 2023-09-24 23:27:21.735204
>> 이번 달은 9월로 가을입니다.
```

Homework

- 로또 - 2
 - 1~45까지 숫자 중 임의로 5개의 숫자를 출력하는 프로그램을 작성하시오. 단, 모든 숫자가 선택 될 확률은 동일해야 하며, 뽑히는 순서는 상관없다.
 - 동일한 숫자가 나오지 않아야 됨
- 윤년 판별
- 2차 방정식 근을 계산
- 삼각형 판별
- datetime 을 이용하여, 오늘의 계절을 구분

Homework

- Code를 zip으로 묶어서 제출
 - File 명 : ch3-1-이름학번-decision.zip
예) ch3-1-김국민20221234-decision.zip
- ecampus 숙제제출 link에 upload
- 제출마감
 - 2023.10.3(화) 13:00
 - 제출 마감 일시까지만 제출 가능. 마감일시 이후 ecampus 숙제제출 링크 자동 close