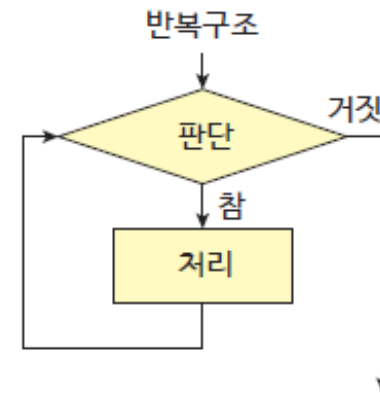
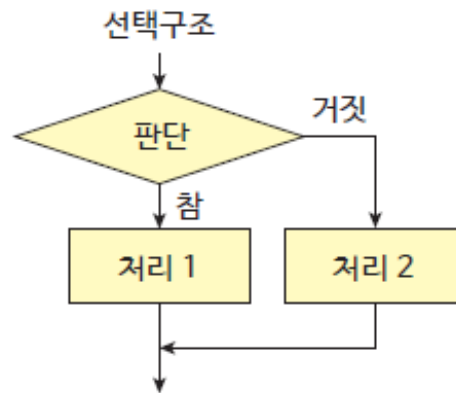
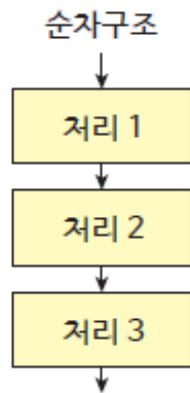


데이터분석 프로그래밍 조건문

임현기

제어 구조

- 순차 구조
- 선택 구조
- 반복 구조



선택 구조

- 선택 구조의 예
 - 미성년자가 아니라면 영화 "킹덤"을 시청할 수 있다
 - 게임에서 철수가 정답을 맞혔으면 철수의 점수를 1만 큼 증가한다
 - 게임 사용자가 외계인 우주선을 맞추었으면 폭발 사운드를 출력한다
 - 파일이 하드 디스크에 없으면 오류 메시지를 출력한다
 - 11:59분이 지나면 수강신청을 할 수 있다

if 문

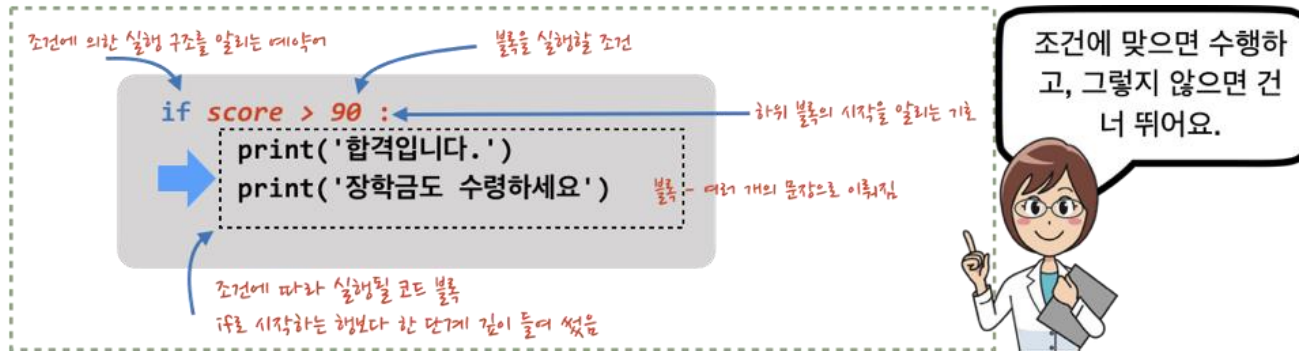
- 조건식: 조건을 만족하는지 아닌지 판정하는 식
 - 조건식은 True 혹은 False로 결정 됨
- :(콜론) 아래 줄은 반드시 들여쓰기 해야 함

```
x = 100
if x > 1:
    print("x는 1보다 큼니다.")
```

```
x는 1보다 큼니다.
```

if 문

- 조건을 만족할 때 2가지를 출력하려면?



- 콜론 아래 문장들은 같은 공백을 가짐
 - 이들 모두는 동일한 블록에 속해있음

if 문

- 들여쓰기 공백을 □로 표현한다면

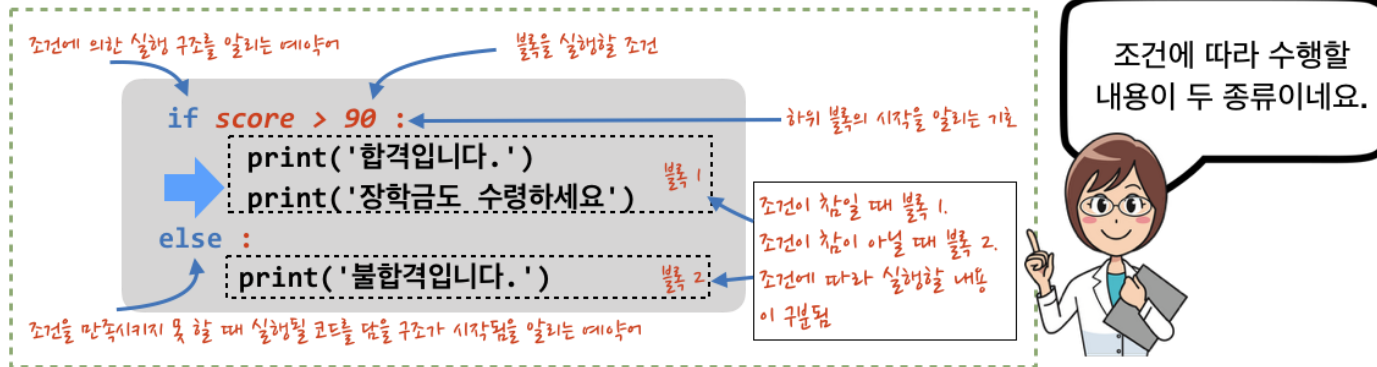
```
score = int(input("점수를 입력하시오: "))
if score >= 90 :           # 조건이 참일 경우 들여쓰기 블록이 모두 실행됨
    □□□□print("축하합니다.")
    □□□□print("합격입니다.")
    □□□□print("장학금도 받을 수 있습니다.")
```

```
score = int(input("점수를 입력하시오: "))
if score >= 90 :           # 주의 : 들여쓰기의 크기가 다를 경우 오류가 출력됨
    □print("축하합니다.")
    □□print("합격입니다.")
    □□□print("장학금도 받을 수 있습니다.")
```

주의 : 들여쓰기의 크기가
달라지면 오류!!

SyntaxError: multiple statements found while compiling a single statement

if-else 문



```
hour = 10  
  
if hour < 12:  
    print('오전입니다.')  
else:  
    print('오후입니다.')
```

if-else 문

- 사용자로부터 성적을 입력 받아서 합격 여부를 판단
 - 사용자의 성적을 입력 받음
 - 입력된 값이 60 이상이면 프로그램은 '합격입니다.'를 출력하고,
 - 그렇지 않으면 '불합격입니다.'를 출력

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

```
성적을 입력하시오: 80  
합격입니다.
```


if-else 문

- 사용자로부터 정수를 입력 받아서 짝수인지 홀수인지를 검사하는 프로그램

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

```
양의 정수를 입력하시오: 10  
짝수입니다.
```

if-else 문

- 만일 임의의 정수를 입력 받은 후 이 검사를 0 이상인 수에 대해 짝수, 홀수를 구분하는 프로그램
 - 음수인 경우
 - 양수인 경우
 - 양수면서 짝수인 경우
 - 양수면서 홀수인 경우

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

```
정수를 입력하시오: 10  
양수입니다.  
짝수입니다.
```

15세 이상만 볼 수 있는 영화가 있다고 하자. 사용자의 나이를 물어 보면 영화를 볼 수 있는지 없는지 여부를 화면에 출력하는 프로그램을 작성해보자.



원하는 결과

나이를 입력하시오: 19
본 영화를 보실 수 있습니다.

나이를 입력하시오: 14
본 영화를 보실 수 없습니다.

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

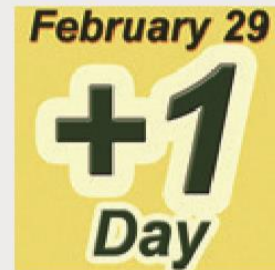
달력은 기본적으로 지구가 태양을 공전하는 시간을 기준으로 작성된다. 하지만 실제로 측정하여 보니 지구가 태양을 완전히 한 바퀴 도는데 걸리는 시간은 365일보다 약 1/4 만큼 더 걸린다. 따라서 매 4년마다 하루 정도 오차가 생기는 셈이다. 이것을 조정하기 위하여 윤년이 생겼다. 입력된 연도가 윤년인지 아닌지를 판단하는 프로그램을 만들어보자.

사용자로부터 연도를 입력받아서 다음과 같은 조건을 검사한다. 윤년은 다음의 조건을 만족해야 한다.

연도가 4로 나누어 떨어지면 윤년이다.

연수가 4로 나누어 떨어져도, 100으로는 나누어 떨어지지 않아야 한다.

연수가 400으로 나누어 떨어지는 해는 무조건 윤년으로 한다.



원하는 결과

연도를 입력하시오: 2012

2012 년은 윤년입니다.

```
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.")
```

이 예제에서는 동전 던지기 게임을 작성해보자. 동전을 던지는 것은 난수를 생성하면 된다. 파이썬에서는 `import random`한 후에 `random.randrange(2)`과 같이 하면 0이나 1을 랜덤하게 생성할 수 있다. 아래의 코드에서 빈칸을 채워보자.

```
import random

print("동전 던지기 게임을 시작합니다.")
coin = random.randrange(2)
if ____:
    print("앞면입니다.")
____:
    print("뒷면입니다.")
print("게임이 종료되었습니다.")
```



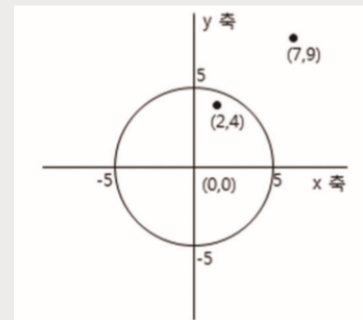
원하는 결과

```
동전 던지기 게임을 시작합니다.
뒷면입니다.
게임이 종료되었습니다.
```

```
import random

print("동전 던지기 게임을 시작합니다.")
coin = random.randrange(2) # randrange(2)는 0 또는 1을 반환함
if coin == 0 :
    print("앞면입니다.")
else :
    print("뒷면입니다.")
print("게임이 종료되었습니다.")
```


중심이 원점 $(0,0)$ 에 있고 반지름이 5인 원이 있다고 가정하자. 사용자로부터 x 와 y 좌표를 입력받은 후, 입력받은 점의 좌표 (x, y) 가 원의 내부에 있으면 '원의 내부에 있음', 원의 외부에 있으면 '원의 외부에 있음'을 출력하는 다음과 같은 프로그램을 작성하시오. (힌트 : 어떤 점이 원 점과의 거리가 5보다 클 경우 원의 외부에 있으며, 5보다 작거나 같을 경우 원의 내부에 있다고 판단할 수 있다. 점 (x, y) 와 원점과의 거리는 $\sqrt{x^2 + y^2}$ 이다)



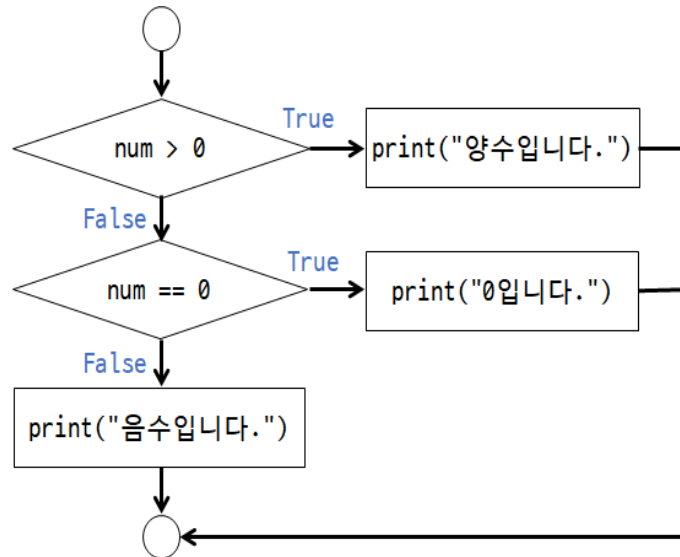
원하는 결과

점의 좌표 x, y 를 입력하시오 : 2 4
 원의 내부에 있음
 점의 좌표 x, y 를 입력하시오 : 7 9
 원의 외부에 있음
 점의 좌표 x, y 를 입력하시오 : 3 6
 원의 외부에 있음

```
x, y = map(float, input('점의 좌표 x, y를 입력하시오 : ').split())
if x*x + y*y > 5*5 :
    print('원의 외부에 있음')
else :
    print('원의 내부에 있음')
```

연속 조건 검사

- if-else 문에서 조건이 거짓일 때 다른 조건 검사
 - 키워드 elif



연속 조건 검사

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

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

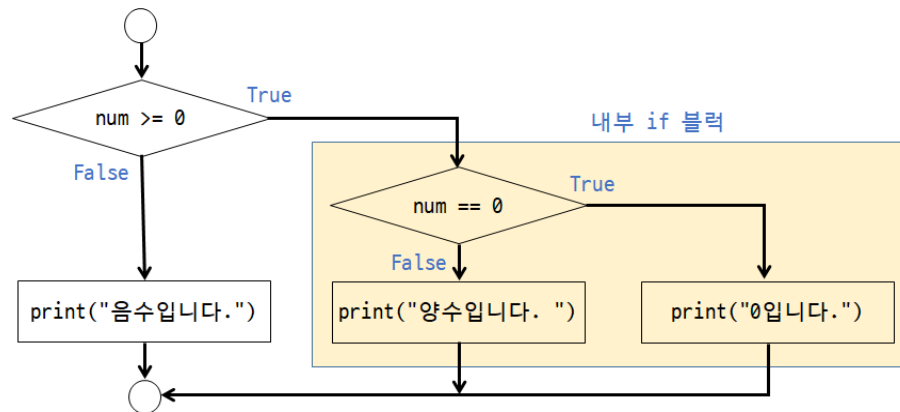
정수를 입력하시오: 0
0입니다.

정수를 입력하시오: 10
양수입니다.

정수를 입력하시오: -10
음수입니다.

중첩 조건문

- if 문 안에 다른 if 문이 들어갈 수 있음



중첩 조건문

```
num = int(input("정수를 입력하시오: "))
if num >= 0:          # 반드시 들여쓰기를 하여 블록을 생성
    if num == 0:      # 블록 내에서 세부적인 조건을 한 번 더 검사
        print("0입니다.")
    else:
        print("양수입니다.")
else:
    print("음수입니다.")
```

정수를 입력하시오: 0
0입니다.

정수를 입력하시오: 10
양수입니다.

정수를 입력하시오: -10
음수입니다.

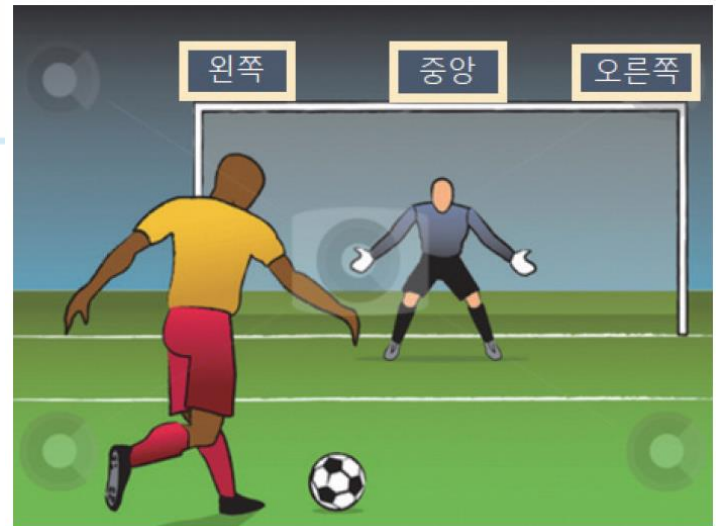


- if-elif-else 문으로 동일한 기능을 하는 프로그램 구현 가능

난수를 이용하여 간단한 축구 게임을 작성하여 보자. 사용자가 컴퓨터를 상대로 페널티킥을 시도한다고 생각하자. 사용자는 다음의 '왼쪽', '중앙', '오른쪽'의 3가지 영역 중에서 하나를 선택하여 페널티킥을 한다. 컴퓨터도 난수를 생성하여 3개의 영역 중에서 하나를 수비한다. 출력은 다음과 같이 나타나도록 하자.

원하는 결과

어디를 공격하시겠어요?(왼쪽, 중앙, 오른쪽) : **왼쪽**
축하합니다!! 공격에 성공하였습니다.
컴퓨터의 수비위치 : 오른쪽



```
import random

n = random.randint(1, 3) # 랜덤하게 1, 2, 3 중 하나의 값을 생성
if n == 1:
    computer_choice = '왼쪽'
elif n == 2:
    computer_choice = '중앙'
else:
    computer_choice = '오른쪽'

user_choice = input('어디를 공격하시겠어요?(왼쪽, 중앙, 오른쪽) : ')
if computer_choice == user_choice:
    print('공격에 실패하셨습니다.')
else :
    print('축하합니다!! 공격에 성공하였습니다.')
print('컴퓨터의 수비위치 :', computer_choice)
```