



# 모두를 위한 파이썬 프로그래밍

5주 데이터 비교판단을 위한 조건문  
(관계연산자, 논리연산자, 조건문)

## ■ 처음 다루었던 연산 프로그램

- ✓ 두수의 연산을 수행하는 프로그램

```
# 연산자 연습
print(10 + 7)
print(10 - 7)
print(10 * 7)
print(10 // 7)    # 정수 나눗셈 계산
print(10 % 7)     # 나머지 계산
```

- ✓ 현실에서는
    - 다양한 연산자의 결과가 필요한 경우는 거의 없다
    - 사용자가 필요한 것을 선택하면, 그 결과만 알려 주면 된다
- 예) 운세를 알려주는 프로그램에서 자신의 운세만 알려주면 된다.

## ■ 순차적으로 진행되는 흐름과 조건적으로 실행하는 흐름이 다른 경우

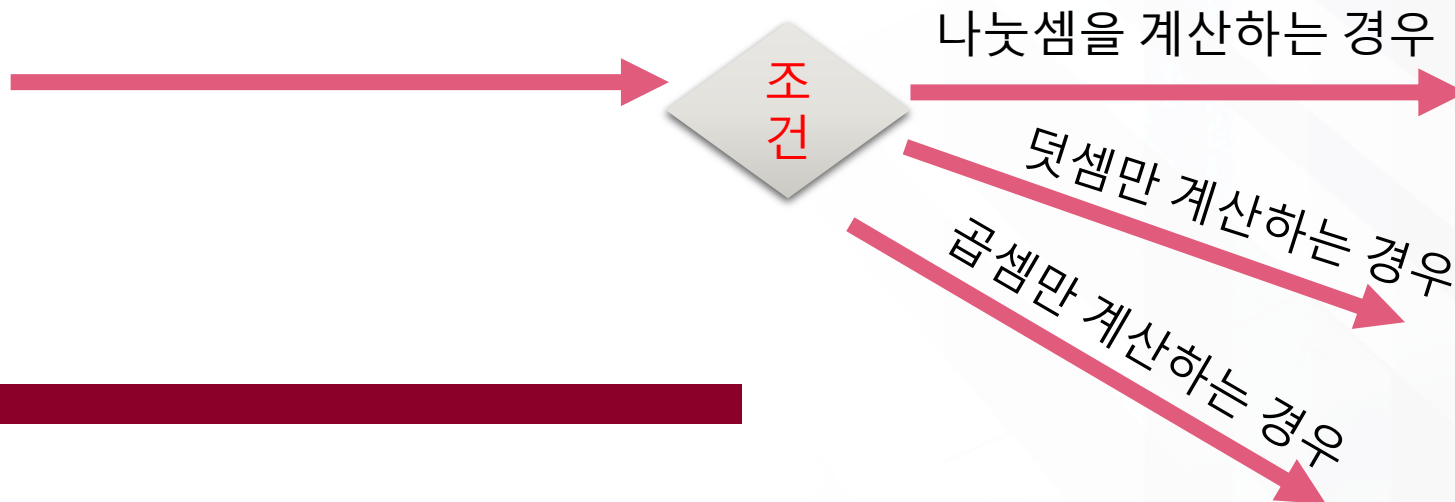
✓ 지난 주차까지 배운 프로그램의 흐름

- 1행부터 ... 차례대로 한 줄씩 모두 수행되었다.



✓ 사용자에게 따라 필요한 조건에 맞는 (필요한)것만 처리해 주는 프로그램

- 모든 행이 처리되지 않고, 필요한 부분만 실행되도록 한다.





## ■ 조건을 어떻게 판별 할 수 있을까?

- ✓ 조건 판단문 → 조건문 → if 조건문 → 결과: True, False

## ■ 값의 같음과 다름을 어떻게 알 수 있을까?

- ✓ 관계(비교) 연산자 이용 → True, False

## ■ 여러 가지 조건들을 동시에 비교 할 수 있을까?

- ✓ 여러 조건으로 구성된 문장 → 논리 연산자 이용 → True, False

## ■ 조건문을 만들 수 있는 문법은 무엇일까?

- ✓ if else

# 관계 연산자(=비교 연산자)



## ■ 두 피연산자의 값을 비교하기 위한 연산자

✓ 결과는 True 또는 False 값으로 나타남 (Boolean)

- True(참)
- False(거짓)

비교 연산자	비교 상태	설명
$x < y$	~보다 작음	x가 y보다 작은지 검사
$x > y$	~보다 큼	x가 y보다 큰지 검사
$x == y$	같음	x와 y의 값이 같은지 검사
$x != y$	같지 않음	x와 y의 값이 같지 않은지 검사
$x >= y$	크거나 같음	x가 y보다 크거나 같은지 검사
$x <= y$	작거나 같음	x가 y보다 작거나 같은지 검사

## ■ 관계 연산자 예제

```
1 A, B = 10, 20
2 print(A == B)
3 print(A != B)
4 print(A > B)
5 print(A < B)
6 print(A >= B)
7 print(A <= B)
8 print(A + 5 < B - 5)
```

## ■ True/False로 결론 내릴 수 있는 두 개의 피연산자를 활용한 논리 연산

- ✓ 논리 연산자 and · or · not문을 사용해 조건을 확장할 수 있다.
  - 빵과 버터를 사오세요 → Buy bread and butter → 둘 다 사야 됨
  - 빵이나 우유를 사오세요 → Buy bread or Milk → 1개 이상만 사면 됨
  - 거짓말이 아니다 → not False → 정말, 진짜(True)

연산자	설명	예시
and	두 값이 모두 참일 경우 True, 그렇지 않을 경우 False	$(7 > 5)$ and $(10 > 5)$ 는 True $(7 > 5)$ and $(10 < 5)$ 는 False
or	두 값 중 하나만 참일 경우 True, 두 값 모두 거짓일 경우 False	$(7 < 5)$ or $(10 > 5)$ 는 True $(7 < 5)$ or $(10 < 5)$ 는 False
not	값을 역으로 반환하여 판단	not $(7 < 5)$ 는 True not $(7 > 5)$ 는 False

## ■ 논리곱 (and)

```
1 A = int(input('수를 입력하세요: '))
2 result = A > 10 and A <= 50
3 print(result)
```

- ✓ 둘 중 모두가 참일 경우에만 참
- ✓ 입력된 값이 9이라면? 10이라면?

논리곱 단순화  
(파이썬에서 가능한 표현)

```
1 A = int(input('수를 입력하세요: '))
2 result = 10 < A <= 50
3 print(result)
```



## ■ 논리합 (OR)

```
1 A = int(input('수를 입력하세요: '))  
2 result = A < 10 or A >= 50  
3 print(result)
```

- ✓ 둘 중 하나만 참이면 결과는 참
- ✓ 입력된 값이 9 일 경우 결과는?
- ✓ 입력된 값이 20 일 경우 결과는?

## ■ 부정 (not)

- ✓ not A : A가 참일 경우 거짓, 거짓일 경우 참

```
1 A = int(input('수를 입력하세요: '))  
2 result = not A  
3 print(result)
```

0일때만 True가 출력된다.

- ✓ 입력된 값이 1이라면? 0이라면? 10이라면? -10이라면?
  - 0 또는 비워있는 변수는 모두 False를 의미하며, 그 외의 값은 모두 참이다.
  - 입력 값이 0 (False)이 입력되면 not 연산자에 의해 반전되어 결과는 True가 출력된다

## ■ 생각해보기

- ✓ "나이가 20대인 경우 참" 을 의미하는 조건식은?
- ✓ "점 (x, y)가 (0, 0) ~ (50, 50)의 사각형 이내에 존재하는 경우 참" 을 의미하는 조건식은?  
 $(x \geq 0) \text{ and } (x \leq 50) \text{ and } (y \geq 0) \text{ and } (y \leq 50)$   
(또는 단순화)  $(0 \leq x \leq 50) \text{ and } (0 \leq y \leq 50)$

### 연산자 우선순위

산술연산자 > 관계연산자 > 논리연산자

# 조건문 개요



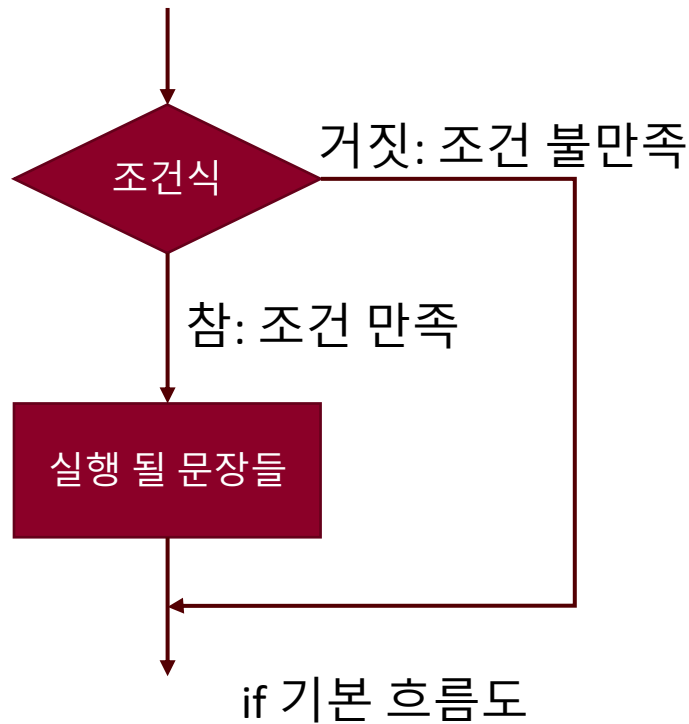
- 조건문(conditional statement): 프로그램 흐름의 분기를 위한 문법
- 관계연산자와 논리연산자의 조합으로 활용
- 조건문의 종류
  - ✓ if
  - ✓ if-else
  - ✓ if-elif-else

# 조건문 if



## ■ 가장 기본적인 조건문

- ✓ 조건식의 결과(True/False)에 따라 구문의 수행 여부를 결정
- ✓ 참(0이 아닌 모든 값)이면 구문을 실행



```
if 조건 :  
    실행 할 문장들
```

조건이 참일 경우 수행됨

기본형 if 구문의 문법

```
1 a = 1  
2 if a == 0:  
3     print('a가 0입니다')  
4 print('프로그램을 종료합니다')
```



# 조건문 if



## ■ 참(True)이란 비어있지 않거나 0이 아닌 모든 값을 말이다

- ✓ None, False, 0, "", (), [], {} 는 0이거나 비워있는 것으로 모두 거짓으로 간주된다.

다음 출력결과가 어떻게 나오는 지 확인해 보자

```
1  if -1:
2      print('-1은 참입니다')
3
4  if 'a':
5      print('a는 참입니다.')
```

### 파이썬 indentation

파이썬에서는 들여쓰기 (indentation) 에 탭문자와 스페이스가 섞여 있으면 에러가 발생한다. 그러나 Pycharm, Visual Code등의 프로그래밍 전용 에디터를 사용하여 코딩하면 자동으로 탭 변환(인덴테이션)을 해 주기 때문에 잘못 작성할 염려가 없다.

# 조건문 if



■ 예제) 유명 홍대 클럽에는 25세 이상은 입장할 수 없는 핫플레이스로 알려져 있다. 나이를 입력 받아 입장 가능 여부를 판별하기 위한 프로그램을 작성하시오

```
*-*-*-*-*-*-*-*|  
HONGDAE CLUB  
*-*-*-*-*-*-*-*
```

```
How old are you? 22  
Welcome to the club
```

```
*-*-*-*-*-*-*-*  
HONGDAE CLUB  
*-*-*-*-*-*-*-*
```

```
How old are you? 26  
No way!!!
```

# 조건문 if



```
1 print('''
2     *-*-*-*-*-*-*-*
3     HONGDAE CLUB
4     *-*-*-*-*-*-*-*
5 ''')
6 age = int(input('How old are you? '))
7 if age < 25:
8     print('Welcome to the club')
9 if age >= 25:
10    print('No way!!!')
```

# 연습문제 1



- 점수를 입력 받아 60점 이상이면 PASS, 60점 미만이면 FAIL를 출력하는 프로그램을 작성하시오

Please enter your Score : 67  
PASS 입니다.

Please enter your Score : 57  
Fail 입니다.

# 연습문제 2



## ■ 점수를 입력 받아 학점을 구하는 프로그램을 작성하시오

✓ 90 ~ 100 → A

✓ 80 ~ 89 → B

✓ 70 ~ 79 → C

✓ 60 ~ 69 → D

✓ 0 ~ 59 → F

Enter your Score : 89

B

Enter your Score : 56

F



# 연습문제 3



- 숫자를 입력받아 홀/짝수 여부를 판별하는 프로그램을 작성하시오

Please input any number : 124

124은 짝수(even) 입니다

Please input any number : 12345

12345은 홀수(odd) 입니다

# if ~ else

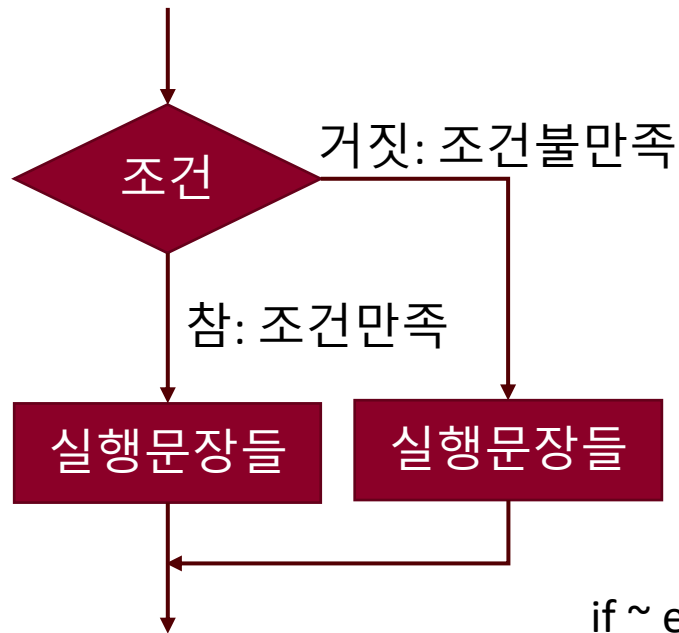


■ 조건식의 결과에 따라 수행 할 구문 결정

■ 참인 경우와 거짓인 경우를 모두 정의

✓ 각 구문은 상호 배타적으로 실행 됨

✓ 참인 경우 참에 해당하는 구문만, 거짓인 경우 거짓에 해당하는 구문만 실행



if ~ else 구문의 흐름도



if ~ else 구문의 문법

# if ~ else



## ■ 조건식이 참인 경우와 거짓인 경우 수행할 문장이 구분되어 있음

# age 변수에 저장되어 있는 값이 25보다 크거나 같다면(25이상이면)

# "당신은 입장할 수 없습니다 " 를 출력하라

```
7  if age < 25:  
8      print('"Welcome to the Club"')  
9  else:  
10     print('Sorry, You are not accepted')
```

# if ~ else



## ■ 홀/짝수 예제

- ✓ 이전 예제에서는 if문으로 작성된 프로그램을 if ~ else 문으로 수정해 보자.

```
1 value = int(input('숫자를 입력하시오 : '))
2 if value % 2 == 0:
3     print('짝수 입니다')
4 if value % 2 == 1:
5     print('홀수 입니다')
```



```
1 value = int(input('숫자를 입력하시오 : '))
2 if value % 2 == 0:
3     print('짝수 입니다')
4 else:
5     print('홀수 입니다')
```

# indentation



## ■ 조건 만족 시 여러 문장을 수행하도록 하려면

- ✓ 여러 문장을 작성하면 된다.

```
1 score = int(input('Enter your Score : '))
2 if score >= 60:
3     print('PASS 입니다.')
4 else:
5     print('Fail 입니다.')
6 print('재수강 하시길 바랍니다.')
```

- ✓ 60점 이하인 경우 Fail을 출력하고, "재수강 하시길 바랍니다."를 출력하고자 한다. 위와 같이 작성한 후 70점을 입력하고 결과를 확인해 보자
- ✓ 이상?한 결과가 나오는 이유는?
  - 해결: indentation을 잘 맞추어 주어야 함



# indentation



## ■ Block

- ✓ 같은 레벨에 존재하는 수행 문장들
- ✓ 인덴테이션(들여쓰기)을 통하여 구분됨
- ✓ 파이썬언어는 다른 언어와 달리 들여쓰기가 매우 중요함

```
1 score = int(input('Enter your Score : '))
2 if score >= 60:
3     print('PASS 입니다.')
4 else:
5     print('Fail 입니다.')
6     print('재수강 하시길 바랍니다.')
```

# if ~ else



## ■ 3항 조건 연산자

- ✓ if/else 구문은 3항 조건 연산자로 동일하게 표현 할 수 있다
- ✓ if condition: A else: B
- ✓ A if condition else B

## ■ 예)

```
if 조건 :
```

```
    print('A')
```

```
else :
```

```
    print('B')
```

=

```
print ('A') if 조건 else print ('B')
```

# if ~ else



## ■ 3항 조건 연산자 예제

```
1 value = int(input('숫자를 입력하시오 : '))
2 if value % 2 == 0:
3     print('짝수 입니다')
4 else:
5     print('홀수 입니다')
```



```
1 value = int(input('숫자를 입력하시오 : '))
2 print('짝수 입니다' if value % 2 == 0 else print('홀수 입니다'))
```



```
1 value = int(input('숫자를 입력하시오 : '))
2 print('짝수 입니다' if value % 2 == 0 else '홀수 입니다')
```

# 중첩된 if



- if else 구문 안에서 또다시 if else 구문을 사용 가능

- ✓ 중첩된 조건 구문

- 예) 입력 받은 숫자가 양수인지, 음수인지, 0인지 판별하시오

숫자를 입력하시오 : **-1**

음수가 입력되었습니다.

```
1 value = int(input('숫자를 입력하시오 : '))
2 if value > 0:
3     print('양수가 입력되었습니다.')
4 else:
5     if value == 0:
6         print('0이 입력되었습니다.')
7     else:
8         print('음수가 입력되었습니다.')
```

## ■사용자로 부터 점수를 입력받아 학점을 출력하는 프로그램을 작성하시오

- ✓ 중첩된 if 구문을 사용할 것
- ✓ 90 - 100 : A
- ✓ 80 - 89 : B
- ✓ 70 - 79 : C
- ✓ 60 - 69 : D
- ✓ 0 - 59 : F

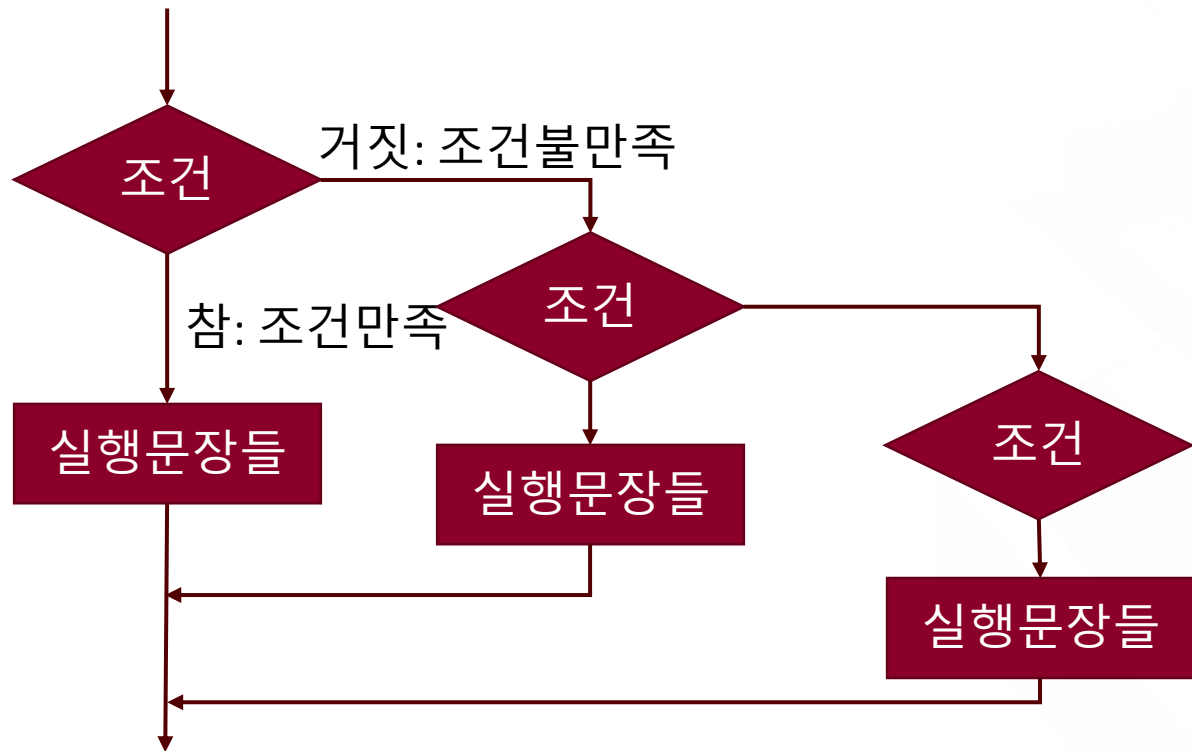
점수를 입력하시오 : 65

학점: D

점수(score)	학점(grade)
98	
37	
16	
86	
71	
63	



- 여러 조건식을 조합하여 분기하기 위한 조건문
- 조건을 만족하는 부분만 수행됨



elif구문의 흐름도

```

if 조건1:
    실행문장들
elif 조건2:
    실행문장들
elif 조건3:
    실행문장들
...
else:
    실행문장들
    
```

## ■ 관계 연산자와 논리 연산자

- ✓ 관계 연산자: 피연산자들간의 비교 (결과는 참 또는 거짓)
- ✓ 논리 연산자: 참 또는 거짓으로 나타나는 피연산자들간의 논리적 연산

## ■ 조건문

- ✓ if 구문: 특정 조건 만족 시 수행)
- ✓ if else 구문: 특정 조건 만족 여부에 따라 다른 내용 수행
- ✓ if elif else 구문: 여러 조건 중 만족하는 경우에 대한 내용 수행
- ✓ if 조건문은 중첩이 가능하다

감사합니다.

