

파이썬 프로그래밍

강의 노트 #04

조건문, 비교연산자, 논리 연산자

선택에 따라 다른 것을 실행

- 프로그램을 작성하다 보면 현재 상황을 확인하고 여러 가지 선택 중 한 가지를 결정해서 실행해야 하는 경우가 발생함
 - 에어컨을 온도에 따라 켤 것인지 끌 것인지 선택
- 여러 가지 선택 중 한 가지를 실행하는 경우도 있음
 - 영화, 뮤지컬 티켓 등은 좌석의 위치에 따라 차등해서 금액이 결정됨

조건식(conditional expression)

□ 조건식

- 조건문을 실행시킬 때 선택을 결정하는 역할
- True, False 또는 이에 준하는 값을 생성하는 표현식
- True 또는 False에 준하는 값이란 불린 자료형은 아니지만, 파이썬에서 True 또는 False로 인식하는 값

비교 연산

- 조건식을 구성하는 가장 일반적인 방법
- 비교(관계) 연산자(Relational operators)
 - 두 개의 피연산자 값을 비교 (같다, 다르다, 크다, 작다 등)

수학	파이썬	설명
$a = b$	<code>a == b</code>	a와 b가 같으면 True
$a \neq b$	<code>a != b</code>	a와 b가 다르면 True
$a > b$	<code>a > b</code>	a가 b보다 크면 True
$a \geq b$	<code>a >= b</code>	a가 b보다 크거나 같으면 True
$a < b$	<code>a < b</code>	a가 b보다 작으면 True
$a \leq b$	<code>a <= b</code>	a가 b보다 작거나 같으면 True

문자열 비교

- 문자열을 비교하는 것은 문자 코드 표를 기준으로 "크다" 또는 "작다"로 얘기할 수 있음
 - 같은 종류의 언어를 비교한다면 사전에 등재된 순서대로 크고 작음을 판단
 - 영문과 한글을 비교하면 영문이 한글보다 작음
 - 영문 대문자와 소문자는 다른 글자로 취급하고, 영문 대문자가 소문자보다 작음
 - 특수기호나 숫자는 영문 알파벳보다 작음

문자열 비교

```
>>> "한글" == "한글"
```

```
True
```

```
>>> "가" < "나" # 같은 언어 비교시 사전 순
```

```
True
```

```
>>> "English" == "English"
```

```
True
```

```
>>> "English" < "Korean" # 같은 언어 비교시 사전 순
```

```
True
```

```
>>> "English" < "한글" # 영문 < 한글
```

```
True
```

```
>>> "English" > "english" # 영문 대문자 < 영문 소문자
```

```
False
```

```
>>> "1234" < "A" # 숫자 < 알파벳
```

```
True
```

숫자 또는 불린 비교

▣ 정수형과 실수형을 섞어서 비교 가능

```
>>> 2 == 2.0
True
>>> 2 < 2.1
True
>>> 2.1 >= 2.3
False
```

▣ 불린 자료형이 같은지 다른지 비교 가능

```
>>> True == True
True
>>> True == False
False
```

숫자 또는 불린 비교

□ 비교 연산식들을 비교할 수도 있음

```
>>> (2 > 3) == (3 > 2) # False == True
False
>>> (3 > 2) == (2 < 3) # True == True
True
```

□ 불린값의 크기 비교도 가능

```
>>> int(True)
1
>>> int(False)
0
>>> True > False
True
>>> True < False
False
```


다른 자료형끼리 비교

- 다른 자료형의 값들을 같은지 비교하면 False, 다른지 확인하면 True를 반환

```
>>> "abc" == (3 > 2) # 문자열과 불린값 비교
False
>>> "abc" != 3.2 # 문자열과 실수값 비교
True
```

- 크기 비교 연산자를 이용해서 다른 자료형을 비교하면 오류 발생

```
>>> "abc" > True
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
TypeError: '>' not supported between instances of 'str' and 'bool'
```

단일 조건문(if문)

□ if 문

- 주어진 조건이 만족될 때 코드 실행

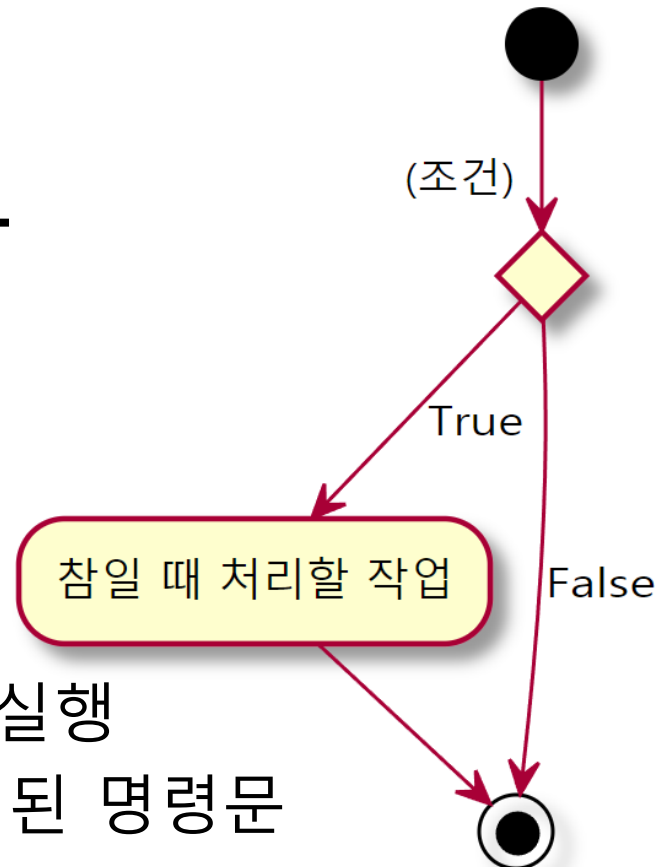
```
if <조건_표현식>:  
    <코드 블록>
```

- 조건_표현식이 만족되면 코드 블록 실행
- 코드 블록은 한 줄 이상의 들여쓰기 된 명령문

```
if True:  
    print("if문 실행")
```

```
>>> if True:  
...   print("if문 실행")  
...  
if문 실행
```

```
>>> if False:  
...   print("if문 실행")  
...
```



단일 조건문(if문)

- 사용자가 입력한 문자열이 "hello"인지 확인

```
s = input("단어 한 개를 입력하세요: ")  
if s == "hello":  
    print("사용자가 입력한 단어가 hello입니다")
```

코드 블록(Code Block)

□ 코드 블록 혹은 블록

- 들여쓰기(indentation)된 한 줄 이상의 코드
- 동일하게 공백을 사용해서 들여쓰기 된 코드
- 들여쓰기 할 때의 공백은 공백 문자 또는 탭 문자를 사용함 (일정하게 한 가지로만 써야 함)
- 들여쓰기 할 때의 공백은 같은 깊이로 해야 함
- 블록은 ':' 다음 줄부터 사용될 수 있음

중첩 if문

- if문의 코드 블록 영역에는 대입문과 조건문을 포함해서 파이썬에서 사용할 수 있는 어떤 명령문도 포함 가능
- 중첩 if문
 - if문의 코드 블록 영역에 또 다른 if문이 포함되는 것

```
if temperature >= 25: # (1) 25도 이상이면
    if humidity >= 70: # (2) 온도 >= 25도이고 습도 >= 70이면
        print("에어컨을 켜야 함") # (3)
    print("end") # (4)
```

조건식에서 다른 자료형이 사용될 때의 의미

□ 숫자

- 0이 아닌 숫자가 조건문에 쓰이면 True, 0이면 False

□ 문자열

- ""이면 False, 문자열에 문자가 포함되어 있으면 True

□ 리스트

- 빈(empty) 리스트이면 False, 리스트에 요소가 있으면 True

□ 튜플(Tuple)

- 빈 튜플이면 False, 튜플에 요소가 있으면 True

□ 딕셔너리(Dictionary)

- 빈 딕셔너리이면 False, 딕셔너리에 요소가 있으면 True

조건식에서 다른 자료형이 사용될 때의 의미

```
>>> n = 0
>>> if n:
...     print("False")
...
>>> s = 'a'
>>> if s:
...     print(s)
...
a
```

실습문제 1

□ 문제

- 사용자로부터 정수 한 개를 입력 받고, num 변수에 저장
- num값이 5보다 크거나 같을 때와 작을 때 표에서 보인 것처럼 화면에 출력

조건	비교 연산	출력 내용
5 이상	<code>num >= 5</code>	5보다 크거나 같습니다
5 미만	<code>num < 5</code>	5보다 작습니다

□ 요구사항

- 사용자가 오류 없이 정수만을 입력할 것이라고 가정

조건문의 부속품 else문

□ 조건문에 종속적인 else문

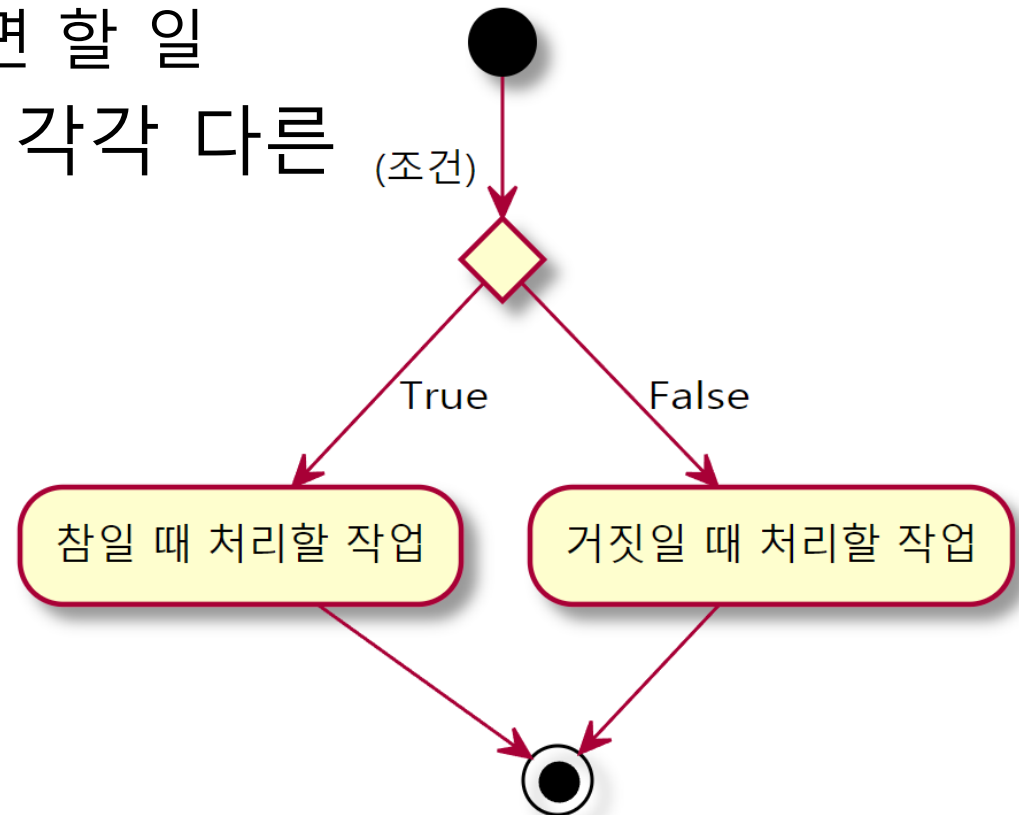
- else문은 조건문에서 조건이 만족되지 않을 때 실행
- if문과 함께 있어야 함
 - else문만 따로 존재할 수 없음

□ else문의 사용 방법

```
if <조건식>:  
    <코드 블록1>  
else:  
    <코드 블록2>
```

조건문의 부속품 else문

- 문제 다시 보기: 5 이상의 정수를 입력 받는 경우와 아닌 경우.
 - 5이상 정수가 입력되면 할 일
 - 5 미만 정수가 입력되면 할 일
- True or False 인 경우 각각 다른 일을 처리해야 함
- if-else문



조건문(if-else문)

□ 예

```
num = 5
if num >= 5:
    print("5보다 크거나 같습니다.")
else:
    print("5보다 작습니다.")
```

□ 사용자가 입력한 문자열이 "hello"인지 아닌지 출력

```
s = input("단어 한 개를 입력하세요: ")
if s == "hello":
    print("사용자가 입력한 단어가 hello입니다")
else:
    print("사용자가 입력한 단어가 hello가 아닙니다")
```

실습문제 2

□ 문제

- 10 이상, 5 이상 10 미만, 5 미만의 정수를 입력 받을 때 적절한 내용을 화면에 출력
- 입력 받은 숫자가 num 변수에 저장되어 있다고 가정

조건	비교 연산	출력 내용
10 이상	$\text{num} \geq 10$	10보다 크거나 같습니다
5 이상 10 미만	$5 \leq \text{num} < 10$	5보다 크거나 같지만 10보다 작습니다
5 미만	$\text{num} < 5$	5보다 작습니다

□ 요구사항

- 사용자가 오류 없이 정수만을 입력할 것이라고 가정

조건문(if-elif-else 문)

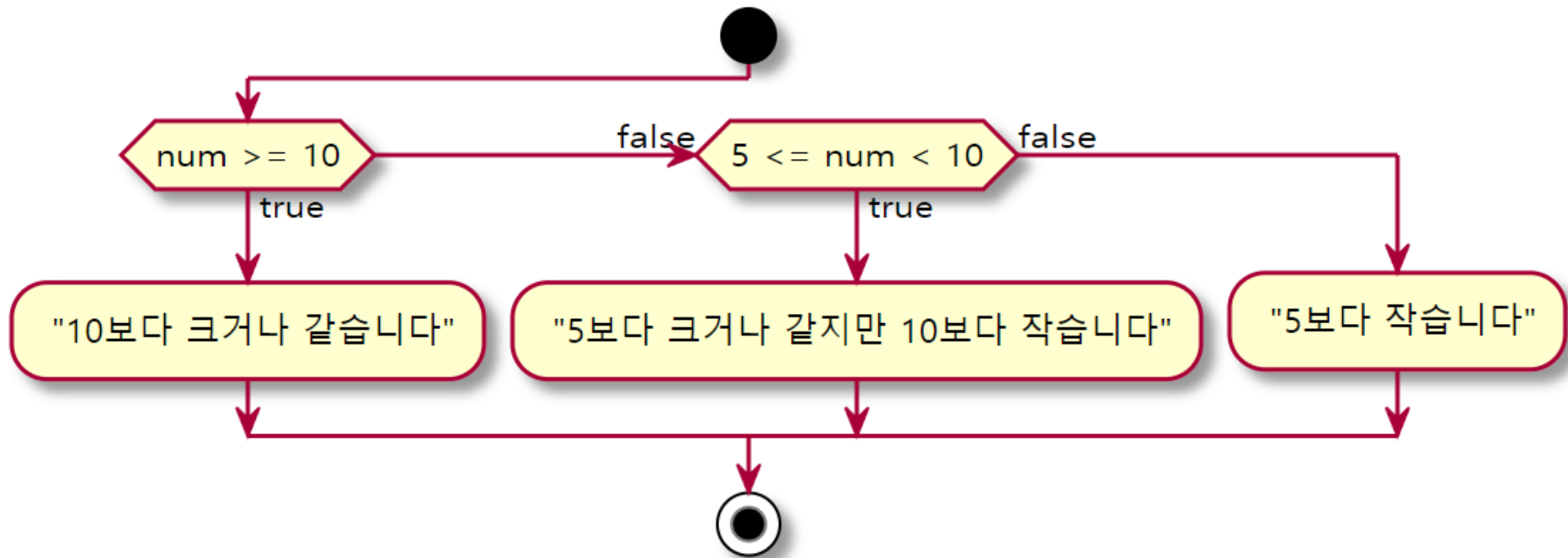
□ 조건문(if-elif-else문)

- 여러 개 중에서 한 개를 선택해야 할 때 사용

```
if <조건표현식1>:  
    <코드 블록1>  
elif <조건표현식2>:      # 생략 가능  
    <코드 블록2>        # 생략 가능  
elif ...  
    ...  
else:                    # 생략 가능  
    <코드 블록n>        # 생략 가능
```

실습문제 2 다시 보기

- 세 가지 조건 중에서 한 가지 선택
 - if-elif-else 구문 사용 가능



실습문제 2 다시 보기

▣ if-elif 구문으로 수정

```
num = 7
if num >= 10:
    print("10보다 크거나 같습니다.")
elif 5 <= num:
    print("5보다 크거나 같지만 10보다 작습니다")
elif num < 5:
    print("5보다 작습니다")
```

실습문제 2 다시 보기

▣ if-elif-else 구문으로 수정

```
num = 7
if num >= 10:
    print("10보다 크거나 같습니다.")
elif 5 <= num:
    print("5보다 크거나 같지만 10보다 작습니다")
else:
    print("5보다 작습니다")
```


조건문(if-elif-else 문)

- 사용자로부터 정수로 된 점수를 입력받고 A(90 이상), B(80 이상 90 미만), C(70 이상 80 미만), D(70 미만)로 성적을 출력
- 첫 번째 버전(if문으로만 처리)

```
score = int(input("점수를 입력하세요: "))
if score >= 90:          # (1)
    print("A")
if 80 <= score < 90:    # (2)
    print("B")
if 70 <= score < 80:    # (3)
    print("C")
if score < 70:          # (4)
    print("D")
```

조건문(if-elif-else 문)

▣ 두 번째 버전(if-elif문으로 처리)

```
if score >= 90:          # (1)
    print("A")
elif 80 <= score < 90: # (2)
    print("B")
elif 70 <= score < 80: # (3)
    print("C")
elif score < 70:        # (4)
    print("D")
```

조건문(if-elif-else 문)

▣ 세 번째 버전(if-elif-else문으로 처리)

```
if score >= 90:  
    print("A")  
elif 80 <= score:  
    print("B")  
elif 70 <= score:  
    print("C")  
else:  
    print("D")
```

실습문제 3

□ 문제

- 2022년에 공연되는 A 뮤지컬의 티켓 가격은 다음 표에 보인 것처럼 좌석의 종류에 따라 다름

좌석 종류	가격
VIP	150000
S	110000
A	90000
B	70000

- 사용자로부터 구매할 좌석의 종류를 입력 받고, 그 좌석에 해당되는 가격을 출력

□ 요구사항

- 표에 없는 좌석이 입력되었다면 잘못 입력했다고 출력

조건식을 조합하는 논리연산자

□ 논리 연산자

- 조건 표현식을 조합해서 코드를 간결하게 만들 수 있음
- and, or, not

피연산자 A	피연산자 B	A and B	A or B	not A
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

□ num 변수가 5 이상 10 미만을 논리 연산자로 표현

```
5 <= num and num < 10
```

조건식을 조합하는 논리연산자

■ 기존 코드 다시 살펴보기

- 온도가 25도 이상이고 습도가 70 이상

```
if temperature >= 25: # 25도 이상이면
    if humidity >= 70: # 온도 >= 25, 습도 >= 70
        print("에어컨을 켜야 함")
```

- 논리 연산자를 이용해서 다시 코드 작성

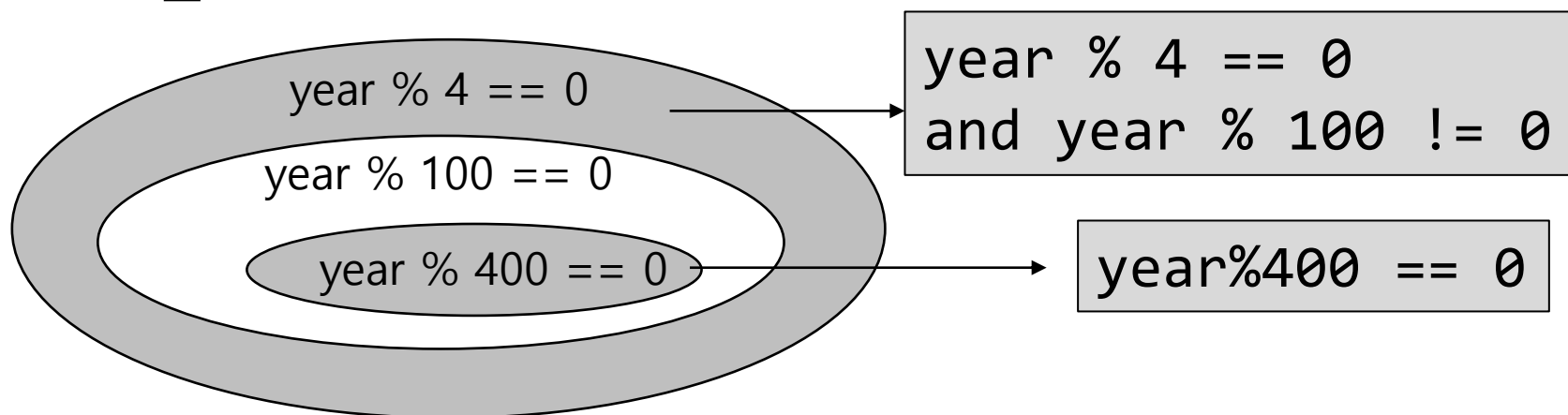
```
if temperature >= 25 and humidity >= 70:
    print("에어컨을 켜야 함")
```

조건식을 조합하는 논리연산자

- 양력에서 평년은 365일로 구성됩니다.
- 항성년(지구가 태양에 대해 한 번 회전하는 데 필요한 시간)의 길이는 실제로 365.2425일,
- 4년에 한 번씩 366일의 "윤년"을 사용함으로써 (짧은)
) 평년 3년으로 인한 오류를 제거합니다. 4로 균등하게 나눌 수 있는 연도는 윤년입니다.
- 예를 들어 1988년, 1992년, 1996년은 윤년입니다.

조건식을 조합하는 논리연산자

- year 변수에 1 이상의 정수를 저장한 후, year가 율년을 나타내는지 확인하는 프로그램 작성
- 윤년인지 판별하는 조건
 - 연도를 4로 나눌 때 나머지가 0이면 윤년. 단 100으로 나눌 때 나머지가 0이면 윤년이 아님
 - 예외적으로 연도를 400으로 나눌 때 나머지가 0이라면 윤년



조건식을 조합하는 논리연산자

□ 윤년을 확인하는 코드

- 두 조건 중 한 가지를 만족시키면 됨(or로 연결)

```
(year % 400 == 0) or (year % 4 == 0 and year % 100 != 0)
```

□ 파이썬 셸에서 실행

```
>>> year = 2000
>>> if (year % 400 == 0) or (year % 4 == 0 and year
% 100 != 0): # (1)
...     print(f"{year}는(은) 윤년입니다")
... else:
...     print(f"{year}는(은) 윤년이 아닙니다")
...
2000는(은) 윤년입니다
```

실습문제 4

□ 문제

- 사용자에게 온도와 습도를 정수로 입력받고, 다음 조건에 맞춰 에어컨을 켜는 프로그램을 작성
 - 온도가 25도 이상이고 습도가 70 이상이면 에어컨을 냉방 모드로 켜
 - 온도가 25도 이상이지만, 습도가 70 미만일 때 제습 모드로 작동
 - 두 조건이 만족되지 않으면 에어컨을 끌 것

□ 요구사항

- 사용자가 정수만 입력할 것이라고 가정
- 실제 에어컨을 켜고 끌 수 없으므로, "냉방 모드로 에어컨 작동", "제습 모드로 에어컨 작동"이라고 출력

단락 평가

□ 단락 평가(short-circuit evaluation)

- 논리 연산식에서 결과를 빨리 평가하기 위해 조건의 일부만 검사하는 방법
- and 연산에서 왼쪽 피연산자의 결과가 False이면 전체 논리 연산의 결과는 False
- or 연산에서 왼쪽 피연산자의 결과가 True이면 전체 논리 연산은 True

단락 평가

- 단락 평가 사용의 두 가지 이유
 - 실행의 효율성
 - 실행 중에 오류가 발생할 수 있는 코드를 논리 연산자의 오른쪽에 놓고 오류를 예방할 수 있는 코드를 왼쪽에 놓으면 오류를 막을 수 있음

```
num = 30
divisor = 0
if divisor != 0 and num / divisor:
    print(f"{num} / {divisor} = { num / divisor }")
```

연산자 우선 순위

산술/비교/논리/대입연산자 우선 순위	연산자
괄호	()
거듭제곱	**
곱셈, 나눗셈, 정수의 나눗셈, 나머지	*, /, //, %
덧셈, 뺄셈	+, -
비교 연산자	<=, <, >, >=
비교 연산자	==, !=
대입 연산자	=, +=, -=, *=, /=, //=, %=, **=
논리 연산자	and, or, not

pass문

□ pass문

- 아무것도 하지 않는 명령문
- 주로 코드 블록에 임시로 코드를 작성하거나 코드 블록을 채워야 하는데 달리 할 일이 없는 경우에 사용

```
if a > 5:  
    print(a)  
else:  
    pass # 아무것도 안하지만, else 부분의 코드  
        # 블록을 구성하고 있다.
```

- 임시로 else 부분을 pass문으로 작성했다가 나중에 다른 코드로 변경할 수 있음

실습문제 5

- 사용자로부터 나이를 입력받아 성인인지 미성년자인지 판별하고 결과를 출력하는 코드를 작성하세요.
- 요구사항
 - 나이가 20살 이상이면 성인

실습문제 6

□ 문제

- 사용자로부터 정수를 입력받아 짝수인지 홀수인지 판별하고 결과를 출력하는 코드를 작성하세요.

□ 요구사항

- 0으로 나누어 떨어지는것으로 판별

실습문제 7

□ 문제

- 사용자로부터 점수를 입력받아 점수에 따라 학점을 부여하고 결과를 출력하는 코드를 작성하세요.

□ 요구사항

- "점수를 입력하세요 (0-100 사이의 정수):"
- 위 문구 활용 등급은 임의로 정의

실습문제 8

□ 문제

- 사용자로부터 섭씨 온도를 입력받아
- 물의 상태(고체, 액체, 기체)를 판별하고
- 결과를 출력하는 코드를 작성하세요.

□ 요구사항

- (0도 이하: 고체,
- 0도 초과 100도 미만: 액체,
- 100도 이상: 기체)

실습문제 9

□ 문제

- 사용자로부터 학년(1-4)을 입력받아 학년에 따라 "초등학생", "중학생", "고등학생", "대학생"을 판별하고 결과를 출력하는 코드를 작성하세요.

□ 요구사항

- 1:초
- 2:중
- 3:고
- 4:대

실습문제 10

□ 문제

- 사용자로부터 세 변의 길이를 입력받아 삼각형이 될 수 있는지 판별하고,
- 어떤 삼각형인지(정삼각형, 이등변삼각형, 일반삼각형) 출력하는 코드를 작성하세요.

□ 요구사항

- 변의 길이 활용

실습문제 11

□ 문제

- 사용자로부터 두 개의 숫자를 입력받아 첫 번째 숫자가 두 번째 숫자의 배수인지 여부와
- 두 숫자의 합이 100 이상인지 판별하여 결과를 출력하는 코드를 작성하세요.

□ 요구사항

- 나머지 활용

실습문제 12

□ 문제

- 사용자로부터 두 개의 문자열을 입력받아 두 문자열이 같거나 그 길이가 같은지 판별하여 결과를 출력하는 코드를 작성하세요.

□ 요구사항

- len() 명령어 활용해보자:문자열 길이

실습문제 13

□ 문제

- 사용자로부터 나이와 운전 경력을 입력받아 나이가 25세 이상이고 운전 경력이 2년 이상이면 보험 가입 가능,
- 그렇지 않으면 보험 가입 불가능을 출력하는 코드를 작성하세요.

□ 요구사항

실습문제 14

□ 문제

- 사용자로부터 구매 금액을 입력받아 100,000원 이상이면 10% 할인,
- 50,000원 이상이면 5% 할인,
- 그 외는 할인이 적용되지 않는 코드를 작성하세요.

□ 요구사항

실습문제 15

□ 문제

- 사용자로부터 패스워드를 입력받아 길이가 8자 이상이고 숫자와 문자를 모두 포함하는지 검증하는 코드를 작성하세요.

□ 요구사항

실습문제 16

□ 문제

- 사용자로부터 나이와 성별을 입력받아 특정 활동을 추천하는 코드를 작성하세요.

□ 요구사항

- 20세 이상의 남성에게는 "헬스장",
- 20세 이상의 여성에게는 "요가", 20세 미만에게는 "공부"를 추천합니다.

실습문제 17

□ 문제

- 사용자로부터 주간 근무 시간을 입력받아 40시간 이상 근무하면 ,
- 초과 근무 시간과 총 급여(시간당 10,000원, 초과 근무 시 1.5배)를 계산하는 코드를 작성하세요.

□ 요구사항

실습문제 18

□ 문제

- 사용자로부터 비밀번호를 입력받아 비밀번호가 8자 이상
- 숫자와 대문자, 소문자를 모두 포함하는지 검증
- 결과를 출력하는 코드를 작성하세요.

□ 요구사항

Geek & geek

