

Python

변수와 연산자

Spring 2025



AI융합학과

Seongbok Baik

sbbaik@dju.kr

00 Text Book



교재명	으뜸 파이썬
저자	박동규, 강영민
출판사	생능출판사
발행년	2024.06.14



학습목표

- 순차적 실행과 조건문, 반복문과 같은 제어문의 차이를 이해하고 활용할 수 있다.
- 조건문의 개념과 사용법에 대해 이해한다.
- 조건식에 대해 알아보고 복잡한 조건식을 구성할 수 있다.
- 블록과 들여쓰기에 대해 이해하고 중첩된 블록을 구성할 수 있다.
- if 문과 if-else 문 등 다양한 조건문에 대해 이해한다.

3.3 if-else 조건문

- 상황 : 시간이 12시가 안 되면 '오전입니다.', 12시 이후면 '오후입니다'를 출력하는 기능

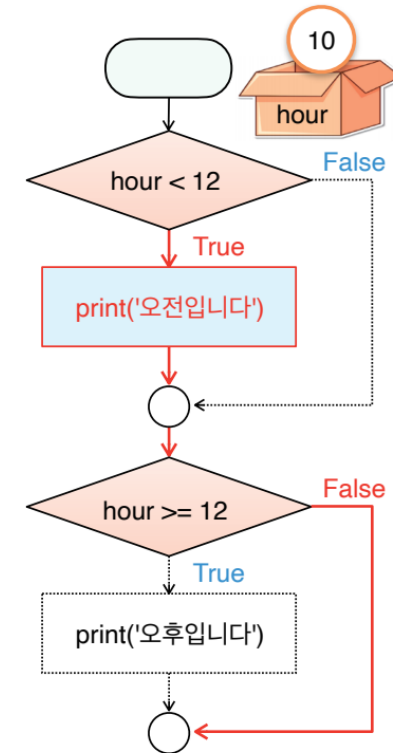


코드 3-8: if 문을 이용한 '오전입니다' 혹은 '오후입니다'의 출력 기능
if_hour_test.py

```
hour = int(input('시간을 입력하세요 :'))
if hour < 12:
    print('오전입니다.')
if hour >= 12:
    print('오후입니다.')
```

실행결과

시간을 입력하세요 : 10
오전입니다.



[그림 3-7] hour 값이 10일 때 코드 if_hour_test.py의 흐름도

3.3 if-else 조건문

- if-else 문을 이용한 출력 : 배타적 관계



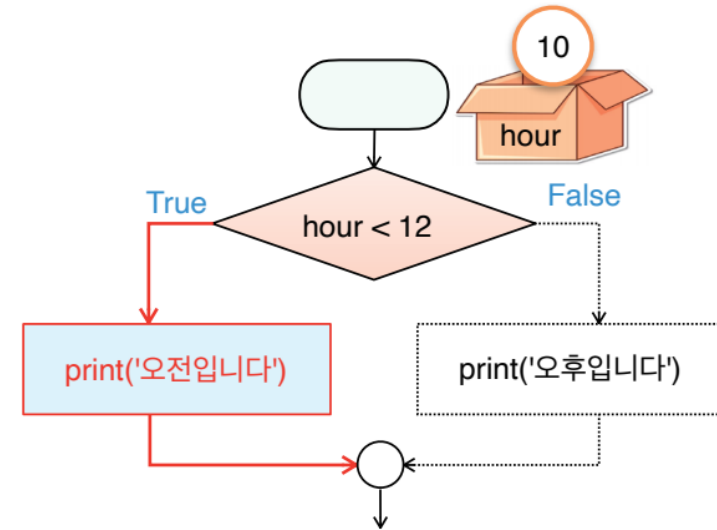
코드 3-9: if-else 문을 이용한 '오전' 혹은 '오후'의 출력 기능
if_else_hour_test.py

```
hour = int(input('시간을 입력하세요 :'))
if hour < 12:
    print('오전입니다.')
else:
    print('오후입니다.')
```

배타적인 관계

실행결과

시간을 입력하세요 : 10
오전입니다.



[그림 3-8] hour 값이 10일 때 if_else_hour_test.py 코드의 흐름도

3.3 if-else 조건문

- 임의의 정수가 음수인지 아닌지 판단하기



코드 3-10: if-else 문을 이용한 '음수' 혹은 '음수 아님'의 출력 기능
if_else_minus_test.py

```
num = int(input('정수를 하나 입력하세요 :'))  
if num < 0:  
    print(num, '은(는) 음수입니다.')  
else:  
    print(num, '은(는) 음수가 아닙니다.')
```

배타적인 관계

실행결과

```
정수를 하나 입력하세요 : -10  
-10 은(는) 음수입니다.
```

3.3 if-else 조건문

- if-else문으로 홀수/짝수를 판별(배타적 관계)



코드 3-11: if-else 문을 이용한 '짝수' 혹은 '홀수'의 출력 기능
if_else_even_test.py

```
num = int(input('양의 정수를 하나 입력하세요 :'))  
if num % 2 == 0:  
    print(num, '은(는) 짝수입니다.')  
else:  
    print(num, '은(는) 홀수입니다.')
```

배타적인 관계

실행결과

양의 정수를 하나 입력하세요 : 10
10 은(는) 짝수입니다.

3.3 if-else 조건문

- if문으로 구성된 블록 내 또 다른 if문이나 if-else 사용



코드 3-12: 외부 if-else 문과 내부 if-else 문의 사용

if_block.py

```
num = 100
if num < 0:
    print(num, '은(는) 음수입니다.')
else:
    print(num, '은(는) 음수가 아닙니다.')
    # 짝수, 홀수는 음수가 아닐 때만 판별함
    if num % 2 == 0:
        print(num, '은(는) 짝수입니다.')
    else:
        print(num, '은(는) 홀수입니다.')
```

실행결과

```
100 은(는) 음수가 아닙니다.
100 은(는) 짝수입니다.
```


3.3 if-else 조건문

- if문으로 구성된 블록 내 또 다른 if문이나 if-else 사용
- 바깥의 if-else문을 **외부 if-else문**이라고 함
 - 외부의 if-else 조건문 블록을 살펴보면 if문은 변수 num의 값이 0보다 작을 때, 즉 음수일 때만 실행
 - else문은 num의 값이 음수가 아닐 때에만 실행
- else문 내의 if-else문을 **내부 if-else문**이라고 함
 - if문은 변수 num을 2로 나누어 나머지가 없을 때, 즉 짝수

num = -100으로 수정한 후의 실행 결과

-100은 음수입력
-100은 음수일 때는 else문 블록이 실행

3.3 if-else 조건문

- if문으로 구성된 블록 내 또 다른 if문이나 if-else 사용



LAB 3-3: if 조건문의 응용

- 게임 사용자의 게임점수(game_score)을 입력 받아서 1,000점 이상이면 '고수입니다.'를 출력하고 1,000점 미만이면 '입문자입니다.'를 출력하는 프로그램을 if-else 문을 이용하여 작성하시오.

게임점수를 입력하시오 : 800

```
game_score = 800
```

입문자입니다.

혹은

게임점수를 입력하시오 : 1300

```
game_score = 1300
```

고수입니다.

3.3 if-else 조건문

- if문으로 구성된 블록 내 또 다른 if문이나 if-else 사용

2. if 문의 복합 조건식을 이용해서 다음과 같은 기능을 수행하는 프로그램을 만들어보자.

1) 우선 '당신은 성인인가요(성인이면 1, 미성년이면 0)' 문을 통해서 성인인지 미성년인지를 구한 다음 미성년이면(0이 입력되면) '당신은 미성년자입니다.'를 출력하고 프로그램을 종료한다.

```
당신은 성인인가요(성인이면 1, 미성년이면 0): 0
당신은 미성년자입니다.
```

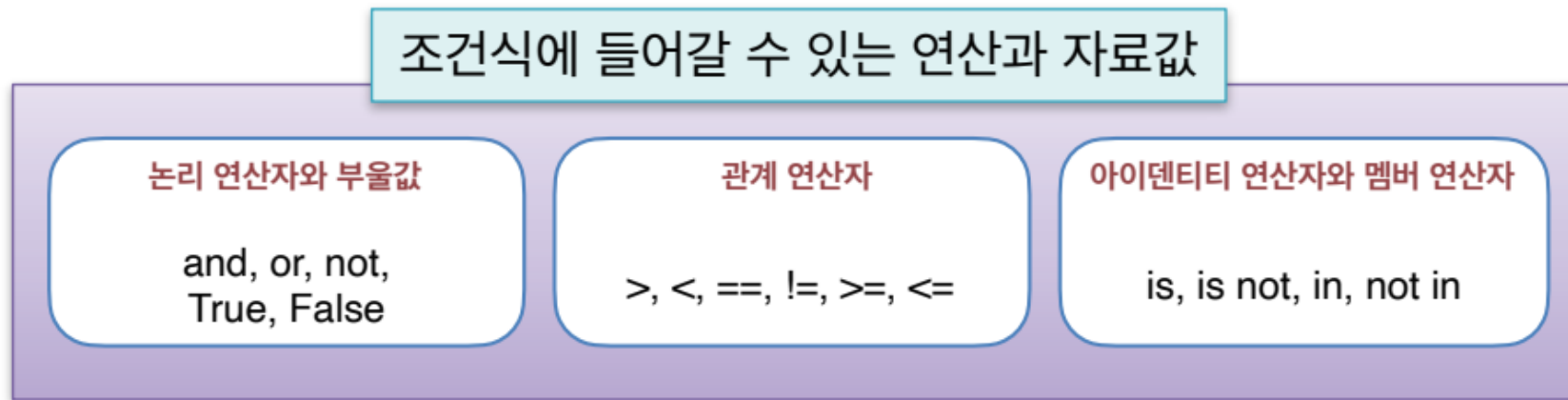
2) 다음으로 성인이면(1이 입력되면) '결혼을 하셨나요(기혼이면 1, 미혼이면 0)' 질문을 통해서 기혼, 미혼을 입력 받아서 '당신은 결혼한 성인입니다.' 혹은 '당신은 결혼하지 않은 성인입니다.'를 다음과 같이 출력하도록 한다.

```
당신은 성인인가요(성인이면 1, 미성년이면 0): 1
결혼을 하셨나요(기혼이면 1, 미혼이면 0): 1
당신은 결혼한 성인입니다.
```

```
당신은 성인인가요(성인이면 1, 미성년이면 0): 1
결혼을 하셨나요(기혼이면 1, 미혼이면 0): 0
당신은 결혼하지 않은 성인입니다.
```

3.3 if-else 조건문

- 3.3.1 복합 조건식
- 더 복잡한 조건을 걸어주기 위해 비교 연산자와 논리 연산자를 조합



[그림 3-9] 조건식에 들어갈 수 있는 연산과 자료값

3.3 if-else 조건문

- 3.3.1 복합 조건식
- 비교 연산자는 연산자 왼쪽의 값과 오른쪽의 값이 해당 연산자의 주어진 조건을 만족할 시 `True` 아니면 `False`

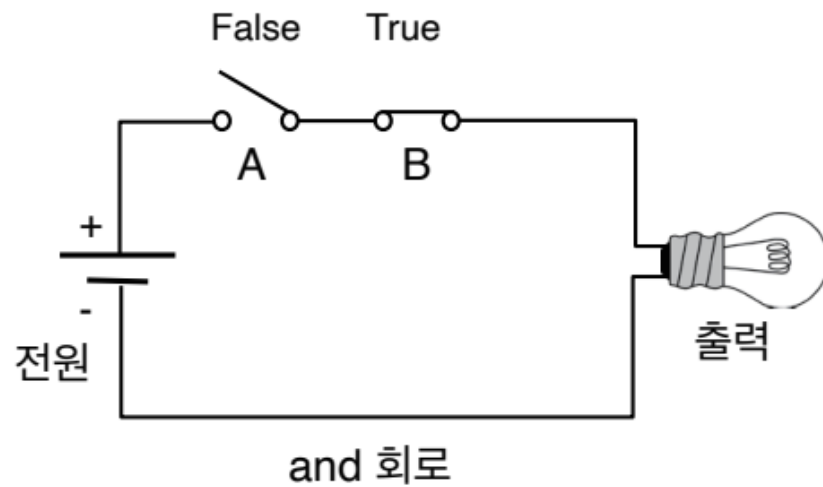


대화창 실습: 조건식 실습

```
>>> 0 < 10          # 조건식이 참, 'True' 출력
True
>>> 15 >= 10        # 조건식이 참, 'True' 출력
True
>>> True or False   # 조건식이 참, 'True' 출력
True
>>> True and False  # 조건식이 거짓, 'False' 출력
False
```

3.3 if-else 조건문

- 논리 연산 and
- 입력 값 중에서 **False** 상태에 영향을 받는 특징이 **있**
으



입력		출력
A	B	
True	True	True
True	False	False
False	True	False
False	False	False

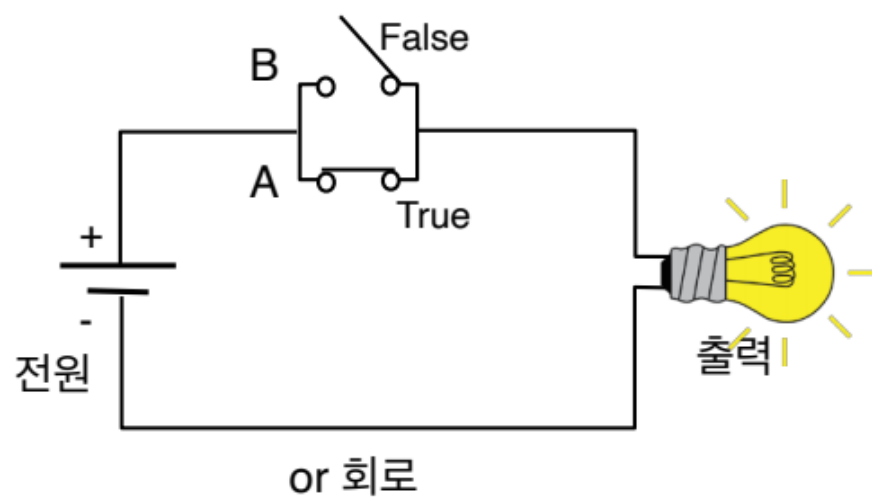
and 회로의 입력과 출력

and 회로는 직렬 회로
이므로 입력된 값들 중
하나라도 False가 있
을 경우 출력은 False
가 됩니다.

[그림 3-10] and 연산을 수행하는 직렬 회로도 and 논리 연산의 결과

3.3 if-else 조건문

- 논리 연산 or
- 출력 값이 **입력 값의 True 상태에 영향을 받음**



입력		출력
A	B	
True	True	True
True	False	True
False	True	True
False	False	False

or 회로의 입력과 출력

or 회로는 병렬 회로이므로 입력된 값들 중 하나라도 True가 있을 경우 출력은 True가 됩니다.

[그림 3-11] or 연산을 수행하는 병렬 회로도 and 논리 연산의 결과

3.3 if-else 조건문

- 변수 a와 b에 입력된 값은 각각 10, 14이므로 두 조건문의 조건식이 모두 참(True)
- 실행 결과는 두 개의 print()문이 모두 실행됨



코드 3-13: and와 or 조건문의 사용법

if_and_or_test.py

```
a = int(input('a를 입력하세요 :'))  
b = int(input('b를 입력하세요 :'))  
if (a % 2 == 0) and (b % 2 == 0): # 첫 번째 조건문  
    print('두 수 모두 짝수입니다.')  
if (a % 2 == 0) or (b % 2 == 0): # 두 번째 조건문  
    print('두 수 중 하나 이상이 짝수입니다.')
```

실행결과

```
a를 입력하세요 : 10  
b를 입력하세요 : 14  
두 수 모두 짝수입니다.  
두 수 중 하나 이상이 짝수입니다.
```


3.3 if-else 조건문

- (b = 13으로 수정하면) 첫 번째 조건문의 조건식을 만족하지 못해 두 번째 조건문 내부의 print문만 실행된다

실행결과

```
a를 입력하세요 : 10
b를 입력하세요 : 13
두 수 중 하나 이상이 짝수입니다.
```

3.3 if-else 조건문



LAB 3-4: 복합 조건식의 이해

1. and 연산자를 사용하여 num 변수가 1과 10 사이의 값을 가지면 True를 출력하는 조건식을 완성하여라.

```
>>> num = 2
```

```
>>> _____
```

```
True
```

2. and 연산자를 사용하여 age가 10보다 크고 19보다 작으면 '청소년입니다.'를 출력하는 조건식을 작성하여라. 그리고 사용자로부터 입력 받은 age에 9와 12를 넣어서 그 결과를 다음과 같이 확인하여라.

```
나이를 입력하세요 : 9
```

위와 같은 경우 아무런 출력을 하지 않으며, 다음과 같이 age에 12가 입력되어야 출력이 된다.

```
나이를 입력하세요 : 12
```

```
청소년입니다.
```

3.3 if-else 조건문

- 3.3.2 복합 조건식으로 윤년 검사하기
- 윤년^{leap year}의 규칙

- 1) 연수가 4로 나누어 떨어지는 해는 윤년으로 한다(예를 들어 1992년, 2004년 등).
- 2) 연수가 1의 조건에 만족함에도 100으로 나누어 떨어지는 해는 평년으로 한다(예를 들어 1900년, 2100년, 2200년, 2300년 등).
- 3) 연수가 400으로 나누어 떨어지는 해는 무조건 윤년으로 한다(예를 들어 2000년, 2400년 등).

3.3 if-else 조건문

- 윤년 판별하기



코드 3-14: 윤년을 판별하기 위한 코드
`if_leapyear_test.py`

```
# 윤년 판별하기
year = int(input('연도를 입력하세요 : '))
is_leap_year = ((year % 4 == 0) and (year % 100 != 0)) or\
               (year % 400 == 0)
print(year, '년은 윤년입니까?', is_leap_year)
```

실행결과

```
연도를 입력하세요 : 2000
2000 년은 윤년입니까? True
```

3.4 if-elif-else 문

- 많은 if문을 사용해 점수대별로 등급을 나누는 학점 산출기

점수	등급
100점 ~ 90점 이상	A
90점 미만 ~ 80점 이상	B
80점 미만 ~ 70점 이상	C
70점 미만 ~ 60점 이상	D
60점 미만	F

3.4 if-elif-else 문



코드 3-15: 'A', 'B', 'C', 'D', 'F' 등급 계산을 위한 if 문
if_grade1.py

```
score = int(input('점수를 입력하세요 : '))
if score >= 90 : # 90 이상인 경우 'A'
    grade = 'A'
if score < 90 and score >= 80 : # 90 미만 80 이상인 경우 'B'
    grade = 'B'
if score < 80 and score >= 70 : # 80 미만 70 이상인 경우 'C'
    grade = 'C'
if score < 70 and score >= 60 : # 70 미만 60 이상인 경우 'D'
    grade = 'D'
if score < 60 : # 60 미만인 경우 'F'
    grade = 'F'
print('당신의 등급은 :', grade)
```

실행결과

```
점수를 입력하세요 : 77
당신의 등급은 : C
```

3.4 if-elif-else 문

- 이전에 살펴본 간단한 if문보다 모든 조건식을 전부 다 검사하는 비효율적인 동작을 함.

• "if문은 'and'와 'or'를 사용할 수 있다. 'and'는 'and'를 사용한다. 'or'는 'or'를 사용한다."

```
if score < 80 and score > 70 :  
# if score < 80 and score >= 70 : 의 오류 코드
```

- 각각의 if문의 의미를 하나하나 파악해야 하기 때문에 오류의 가능성이 높아짐
- 이를 해결하기 위하여 다음과 같이 if-else 문을 적용

3.4 if-elif-else 문



코드 3-16: 'A', 'B', 'C', 'D', 'F' 등급 계산을 위한 복합 if 문
if_grade2.py

```
score = int(input('점수를 입력하세요 : '))
if score >= 90:          # 90 이상인 경우 'A'
    grade = 'A'
else:
    if score >= 80 : # 90 미만 80 이상인 경우 'B'
        grade = 'B'
    else:
        if score >= 70: # 80 미만 70 이상인 경우 'C'
            grade = 'C'
        else:
            if score >= 60: # 70 미만 60 이상인 경우 'D'
                grade = 'D'
            else: # 60 미만인 경우 'F'
                grade = 'F'

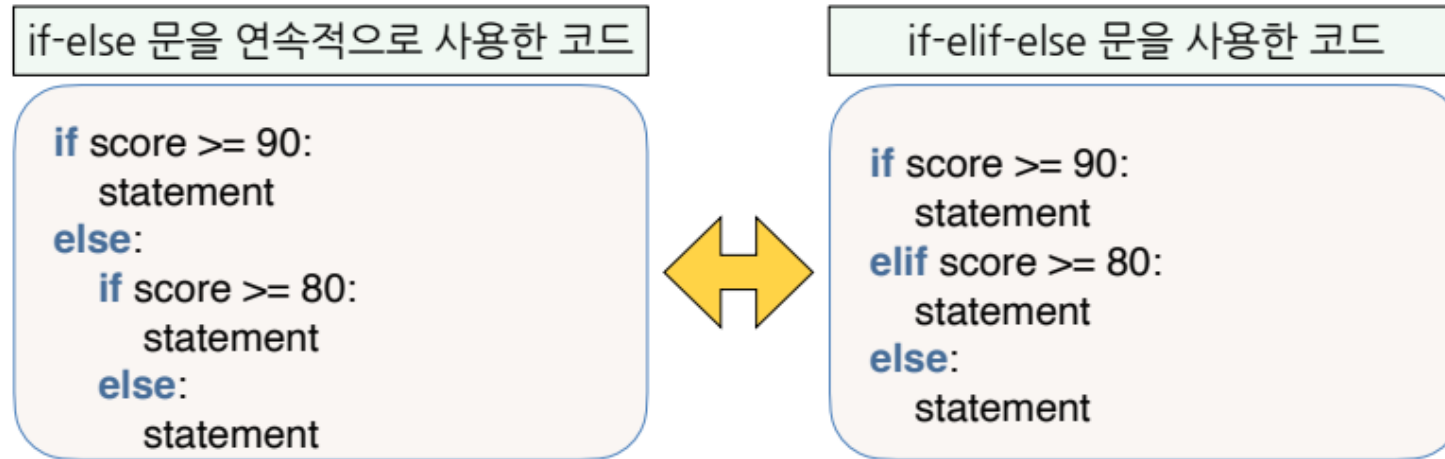
print('당신의 등급은 :', grade)
```


3.4 if-elif-else 문

- 이전의 if문으로만 구성되어 있던 [코드 3-15]보다는 읽기가 편해짐
- 오류의 가능성도 이전에 비해서 줄어듦
- if-else가 조건을 2개밖에 나타낼 수밖에 없기 때문에 가독성은 여전히 떨어짐
- 조건이 여러 개인 경우 if문에서 else문까지 가기 전에 조건을 더 걸어줄 수는 없을까?
-> **elif문 사용하기**

3.4 if-elif-else 문

- if-else문과 elif문의 비교

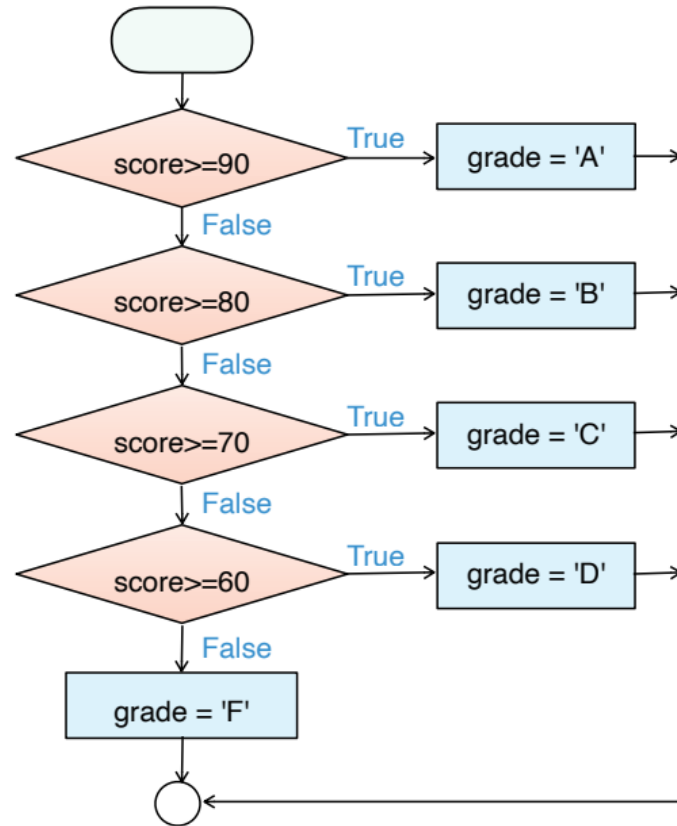


[그림 3-12] if-else 문을 연속적으로 사용한 코드(좌)와 if-elif-else 문을 사용한 코드(우)

- 왼쪽과 오른쪽의 코드는 동일
- 오른쪽의 코드가 들여쓰기도 더 적게 하고 줄의 수

3.4 if-elif-else 문

- if-elif-else 문의 실행 흐름도



[그림 3-13] if-elif-else 문의 실행 흐름도

3.4 if-elif-else 문



코드 3-17: if-elif-else 문으로 구성된 등급계산기

if_elif_grade.py

```
score = int(input('점수를 입력하세요 : '))
if score >= 90: # 90 이상인 경우 'A'
    grade = 'A'
elif score >= 80: # 'A'가 아닌 경우, 80 이상이면 'B'
    grade = 'B'
elif score >= 70: # 'B'도 아닌 경우, 70 이상이면 'C'
    grade = 'C'
elif score >= 60: # 'C'도 아닌 경우, 60 이상이면 'D'
    grade = 'D'
else: # 그 외의 경우 'F'
    grade = 'F'
print('당신의 등급은 :', grade)
```

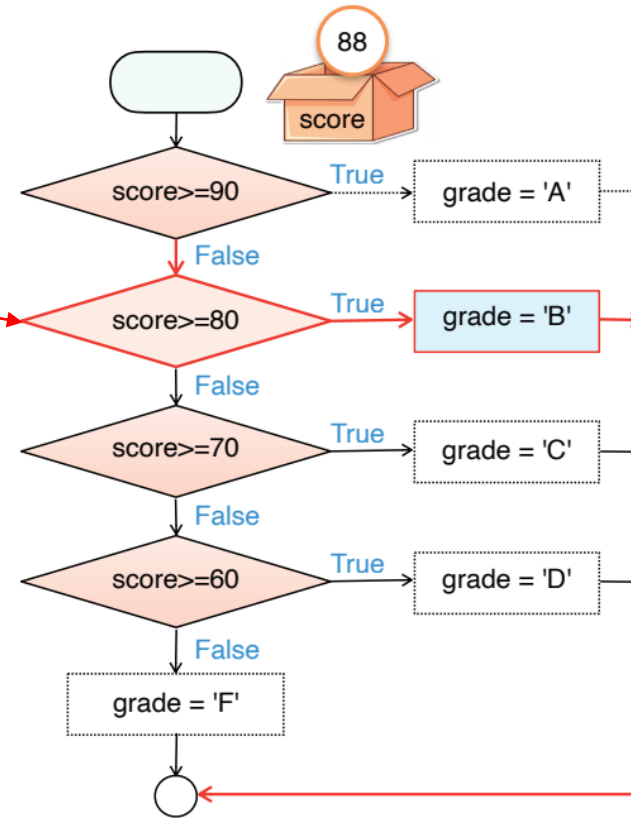
실행결과

```
점수를 입력하세요 : 88
당신의 등급은 : B
```

3.4 if-elif-else 문

- if-elif-else문의 구조

score 변수의 값이 88이면 다음과 같은 실행 흐름을 가지게 된다



[그림 3-14] score 값이 88일 때 if-elif-else 문의 수행 흐름

3.4 if-elif-else 문



LAB 3-5: if-elif-else 문을 사용한 다중 조건식

1. 사용자로부터 자동차의 속도(speed)를 km/h 단위의 정수로 입력 받도록 하자. 자동차의 속도가 100km/h 이상이면 '고속', 100km/h 미만 60km/h 이상이면 '중속', 60km/h 미만이면 '저속'을 출력하는 프로그램을 if-elif-else 문을 이용하여 작성하여라.

자동차의 속도를 입력하세요(단위 : km/h): 13
저속

혹은

자동차의 속도를 입력하세요(단위 : km/h): 130
고속

3.4 if-elif-else 문

2. 미세먼지 농도 예보 기준이 다음과 같을 때 미세먼지 농도를 정수 값으로 입력 받아 ' 좋음', '보통', '나쁨', '매우 나쁨'을 출력하는 코드를 작성하여라(단위 $\mu\text{g}/\text{m}^3$).

좋음	보통	나쁨	매우 나쁨
0 ~ 15	16 ~ 35	36 ~ 75	76 이상

미세먼지 농도를 입력하세요(단위 : microgram/m³) : 13
 좋음

혹은

미세먼지 농도를 입력하세요(단위 : microgram/m³) : 33
 보통

Leistung ist nicht alles / Keinen Studierenden zurücklassen

