

Chapter 5

조건문

임경태



Chapter 05

조건문



Preview



학습목표

- 기본 제어구조 3가지에 대해서 이해합니다.
- 비교 연산자와 논리 연산자의 활용을 정확히 이해합니다.
- 기본적인 if문을 사용한 코드를 작성합니다.
- if~else문 및 elif문에 대한 활용 코드를 작성합니다.

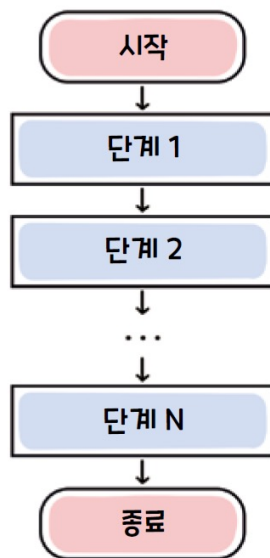
Section 01

조건문의 기초 다지기

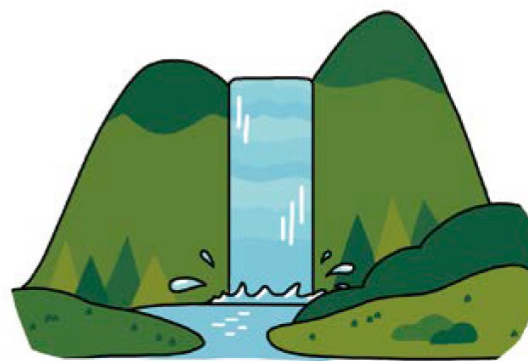


■ 순차 구조

- 순차적(Sequential)으로 코드가 실행됨
- 물이 위에서 아래로 흐르듯이 차례대로 흘러가는 것과 같음
- [시작]에서부터 [종료]까지 각 단계가 순서 그대로 실행되는 구조
- 가장 단순한 구조이기도 하지만, 실제로 많이 사용되는 구조이기도 함



(a) 순차 구조의 흐름



(b) 물이 흐르는 순차 구조

그림 5-1 순차 구조



■ 순차 구조의 예시

- 마트에서 장보기

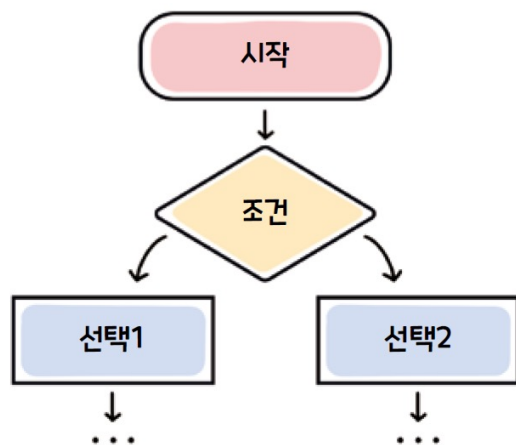


그림 5-2 순차 구조의 예(마트에서 장보기)



■ 선택 구조

- 두 가지의 경우 중에서 어느 하나를 선택한 방향으로 코드가 실행됨
- 길에서 갈림길을 만나면 어느 한쪽으로만 가야 하는 것과 같음



(a) 선택 구조의 흐름

그림 5-3 선택 구조



(b) 갈림길의 선택 구조



■ 선택 구조의 예시

■ 외출 준비하기

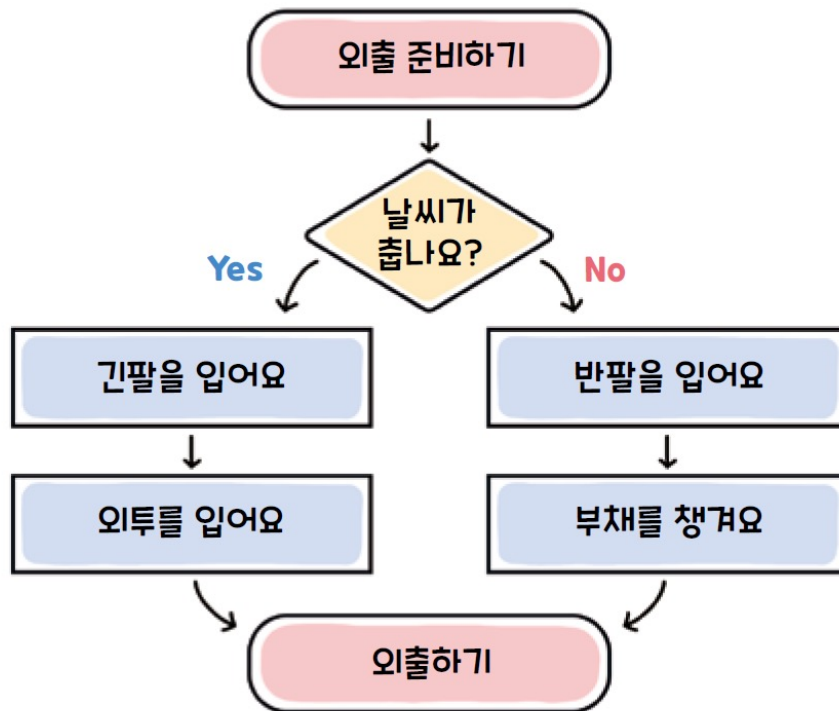
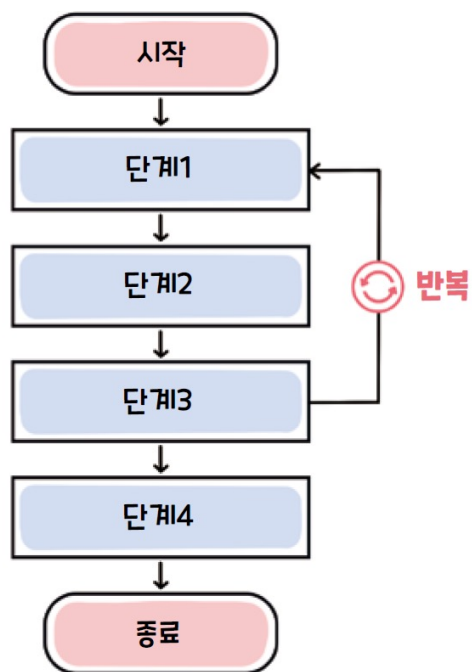


그림 5-4 선택 구조의 예(외출 준비하기)



■ 반복 구조

- 동일한 코드, 특정한 단계가 여러 번 반복해서 실행됨
- 강아지가 꼬리를 물려고 계속 동일한 자리를 반복해서 도는 것과 같음



(a) 반복 구조의 흐름

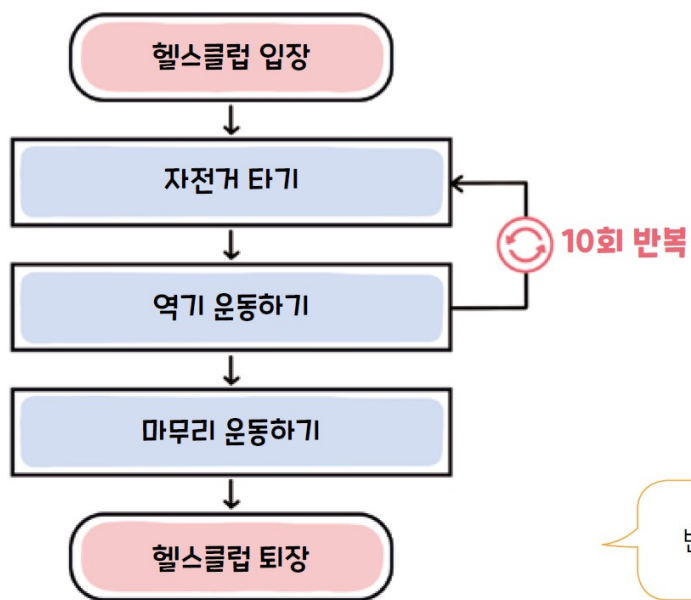
그림 5-5 반복 구조



(b) 같은 행동을 반복하는 구조



- 반복 구조의 예시
 - 헬스장에서 운동하기



반복 구조는 6장에서 자세히 학습합니다.

그림 5-6 반복 구조의 예(헬스클럽에서 운동하기)



확인문제

다음 빈칸에 들어갈 내용으로 올바른 것을 넣으시오.

- 2가지 중 하나의 흐름으로 실행되는 구조는 이다.
- 순서대로만 실행되는 구조는 이다.

① 순차 구조 - 선택 구조

② 순차 구조 - 반복 구조

③ 선택 구조 - 순차 구조

④ 반복 구조 - 순차 구조

정답

Click!



- 실생활에서 두 가지 중 하나를 선택하는 경우
 - 운전면허 필기시험에서 70점 이상이면 합격, 70점 미만이면 불합격인 경우

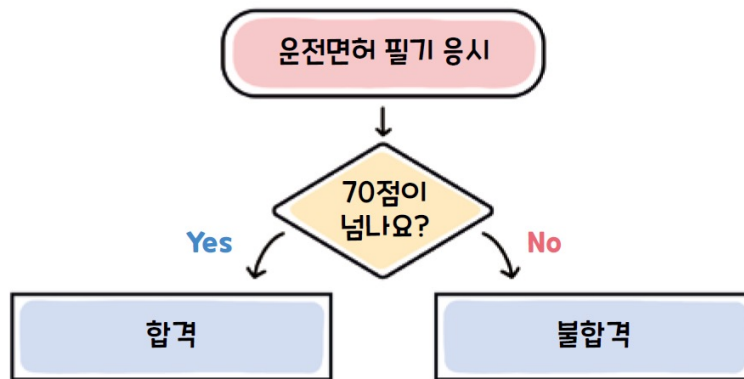
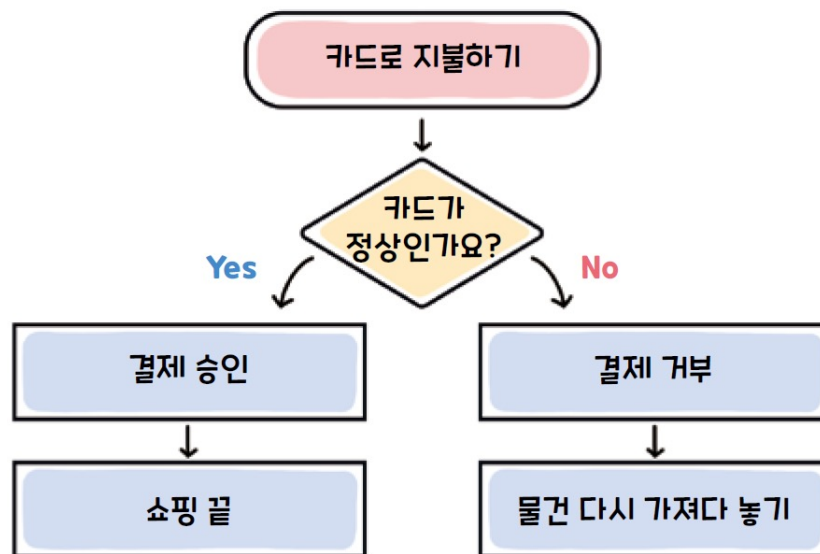


그림 5-7 운전면허 필기시험 흐름도



- 실생활에서 두 가지 중 하나를 선택하는 경우
 - 마트에서 카드가 정상 결제되는 경우와 그렇지 않은 경우



Section 02

조건문 알아보기

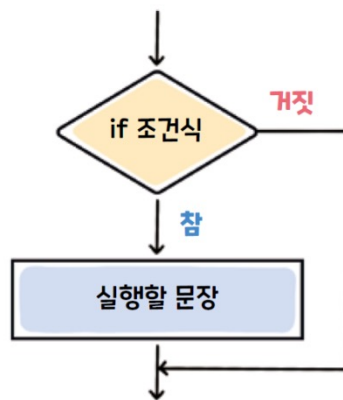


■ if문

- '만약에 ~ 라면'으로 해석
- 참일 때는 무엇을 실행하고, 거짓일 때는 아무것도 하지 않는 가장 단순한 형태의 조건문



그림 5-8 기본 if문의 형식



if문을 순서도로 표현할 때는 주로 마름모 형태로 그립니다.



■ if문의 사용형식

- 'if 조건식 : '에서 조건식이 참이라면 실행할 문장이 실행됨.
거짓이라면 아무것도 실행하지 않고 프로그램을 종료함

[코드 5-1]

```
num = 99  
if num < 100 :  
    print("100보다 작습니다.")
```

[실행결과]

100보다 작습니다.

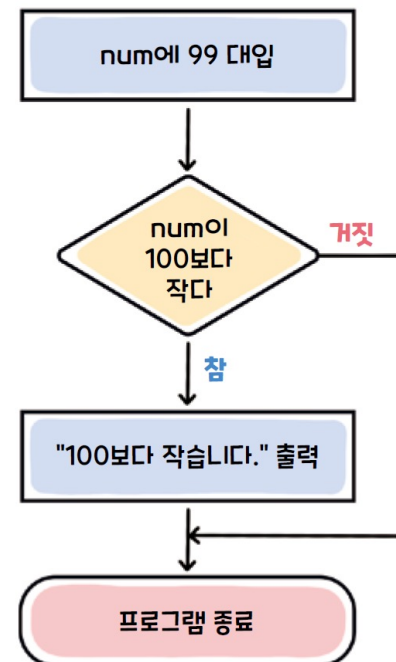
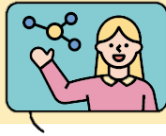


그림 5-9 [코드 5-1]의 실행 과정



- 조건이 참일 때 실행할 문장이 2개인 경우

[코드 5-2]

```
num = 200
if num > 100 :
    print("100보다 ")
    print("큽니다.")
    print("프로그램 끝")
```

[실행결과]

```
100보다
큽니다.
프로그램 끝
```



하나 더 알기 ✓

들여쓰기의 중요성

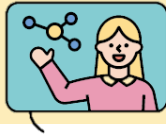
파이썬에서는 들여쓰기가 매우 중요합니다. if문 다음에 실행할 문장은 if와 같은 줄이 아닌 들여쓰기를 해야 합니다. 들여쓰기를 할 때는 `Tab` 보다는 `Spacebar` 4개 정도로 들여쓰기 하는 것을 권장합니다.

그래서 파이썬 스크립트 모드에서 들여쓰기는 `Spacebar` 4개로 자동 지정되어 있습니다. 또한 대화형 모드에서는 실행할 문장이 모두 끝나면 `Enter`를 두 번 눌러야 if문이 끝나는 것으로 간주하고 실행합니다.

한 가지 더 주의할 점은 들여쓰기는 모두 4칸으로 맞춰야 한다는 점입니다. 만약 하나는 3칸, 하나는 4칸이면 실행할 때 오류가 발생합니다. 만약 [코드 5-2]의 4, 5행을 다음과 같이 수정하면 실행할 때 오류가 발생합니다.

```
공백 4칸
print("100보다 ")
공백 3칸
print("큘니다.")
```





확인문제

다음 단순 if문은 오류가 발생한다. 그 이유는 무엇인가?

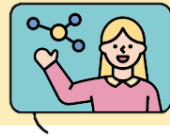
```
number = 100
if number == 10 :
    print("10입니다")

print("프로그램 종료")
```

- ① number 변수 이름에 문제가 있다.
- ② 조건식 if number == 10에 문제가 있다.
- ③ if문 내부의 들여쓰기에 문제가 있다.
- ④ 조건식 다음에 빈 줄이 들어가 있다.

정답

Click!



■ if~else문

- 거짓일 때 실행해야 할 문장이 따로 있을 때 사용함
- 조건식이 참이라면 문장 1을 실행하고, 그렇지 않으면 문장 2를 실행

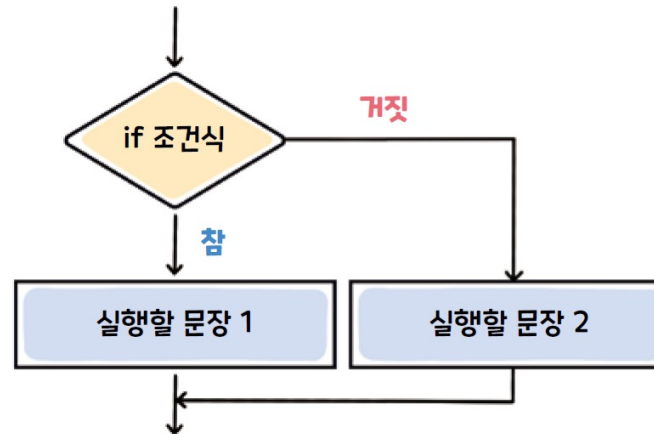
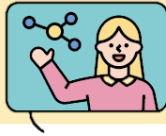


그림 5-10 if~else문의 형식



■ if~else문의 사용 형식

[코드 5-3]

```
num = 200
if num < 100 :
    print("100보다 작군요.")
else :
    print("100보다 크군요.")
```

[실행결과]

100보다 크군요.

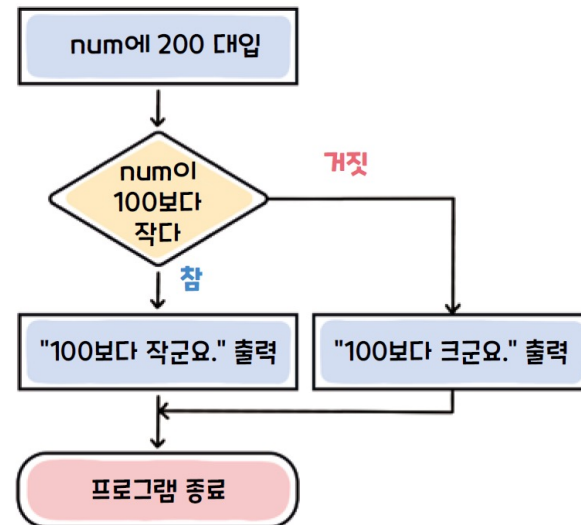


그림 5-11 [코드 5-3]의 실행 과정



■ if~else문의 사용 형식

- 참일 때와, 거짓일 때 수행할 문장이 여러 개인 경우

[코드 5-4]

```
num = 200

if num < 100 :
    print("100보다 작군요.")
    print("여기는 참입니다.")
else :
    print("100보다 크군요.")
    print("여기는 거짓입니다.")

print("프로그램 끝!")
```

[실행결과]

100보다 크군요.
여기는 거짓입니다.
프로그램 끝!

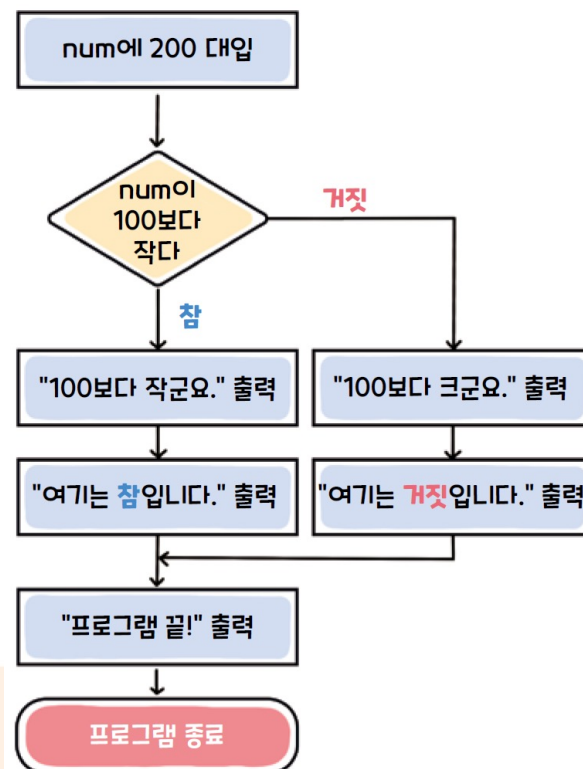


그림 5-12 [코드 5-4]의 실행 과정



■ 숫자가 짝수인지 홀수인지를 알아내는 프로그램

- 짝수: 2로 나눴을 때 나머지 값이 0
- 홀수: 2로 나눴을 때 나머지 값이 1

[코드 5-5]

```
num = int(input("숫자를 입력 ==> "))  
  
if num % 2 == 0 :  
    print("짝수입니다.")  
else :  
    print("홀수입니다.")
```

[실행결과]

숫자를 입력 ==> 1773 ● — 사용자 입력
홀수입니다.



확인문제

다음은 홀수, 짝수를 구분하는 코드이다. 빈칸에 들어갈 문장을 고르시오.

```
number = int(input("숫자 ==> "))
```

```
if  :
```

```
    print("홀수입니다.")
```

```
else :
```

```
    print("짝수입니다.")
```

- ① `number % 2 >= 0`
- ② `number % 2 <= 0`
- ③ `number % 2 == 0`
- ④ `number % 2 != 0`

정답

Click!



■ 중첩 if문

- 조건을 검사하는 과정이 2번 이상인 경우
- 서울에 사는 학생 중에서 25세 이상인 학생이 몇 명인지를 구하는 경우
 - 우선 학생의 주소가 서울인지 아닌지를 확인해야 함
 - 그리고 주소가 서울인 학생 중에서 25세 이상 여부를 구별해야 함
- if문 안에 또 다른 if문이 있는 형태
 - 하지만 조건식이 여러 개 겹쳐지면 헷갈릴 수 있으므로 주의해야 함

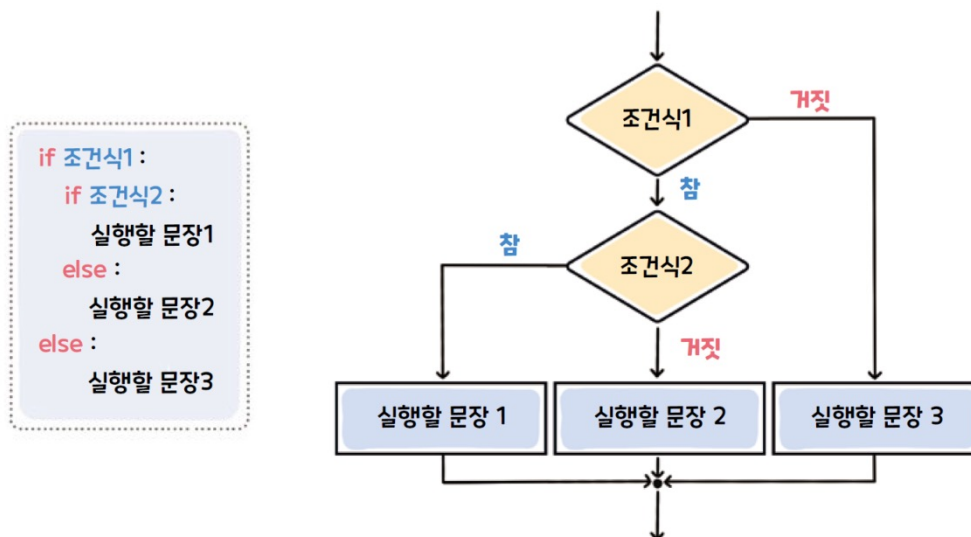


그림 5-13 중첩 if문의 형식



- '100보다 작다', '100에서 1000사이다', '1000보다 크다'로 구분하는 코드

[코드 5-6]

```
num = int(input("숫자를 입력 ==> "))

if num > 100 :
    if num < 1000 :
        print("100보다 크고 1000보다 작군요.")
    else :
        print("와~ 1000보다 크군요.")
else :
    print("음~ 100보다 작군요.")
```

[실행결과]

숫자를 입력 ==> 150 ● — 사용자 입력
100보다 크고 1000보다 작군요.



- 90점 이상은 A, 80점 이상은 B, 70점 이상은 C, 60점 이상은 D, 그 외에는 F로 처리하는 프로그램

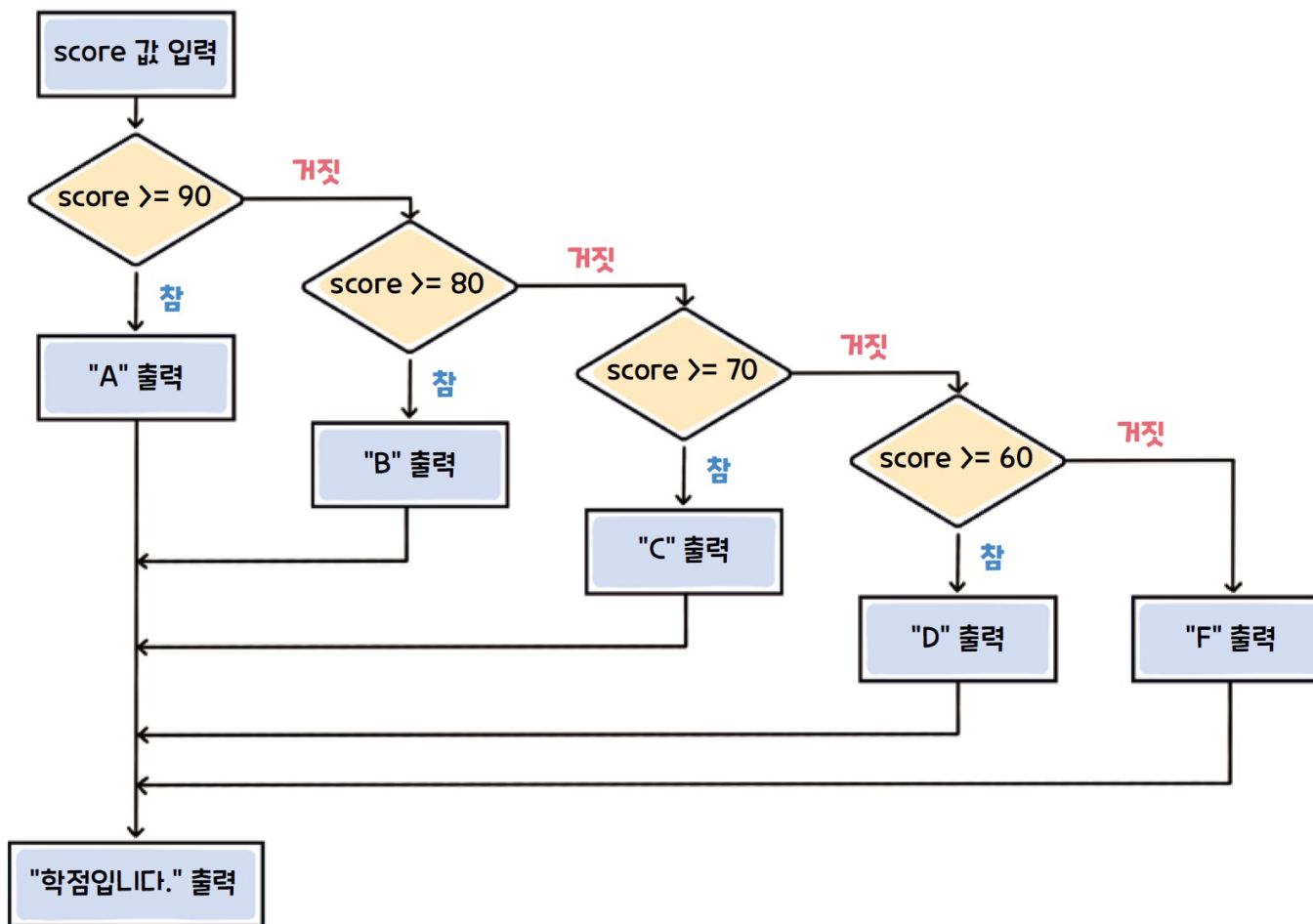


그림 5-14 [코드 5-7]의 실행 과정



- 90점 이상은 A, 80점 이상은 B, 70점 이상은 C, 60점 이상은 D, 그 외에는 F로 처리하는 프로그램

[코드 5-7]

```
score = int(input("점수를 입력 ==> "))

if score >= 90 :
    print("A", end='')
else :
    if score >= 80 :
        print("B", end='')
    else :
        if score >= 70 :
            print("C", end='')
        else :
            if score >= 60 :
                print("D", end='')
            else :
                print("F", end='')
print("학점입니다.")
```

[실행결과]

점수를 입력 ==> 77 ● 사용자 입력
C학점입니다.



확인문제

다음은 10보다 크다/작다/같다를 구분하는 코드이다. 빈칸에 들어갈 문장을 고르시오.

```
number = int(input("숫자 ==> "))

if  :
    if number > 10 :
        print("10보다 크네요.")
    else :
        print("10이네요.")
else :
    print("10보다 작네요.")
```

- ① number == 10
- ③ number >= 10

- ② number <= 10
- ④ number != 10

정답

Click!



■ elif문

- if~else문 안에 다시 if~else문을 사용하면 복잡해 보임
- 그래서 중첩 if문을 사용할 때는 중간의 if~else문을 elif문으로 줄여서 사용할 수 있음
- if~elif~else문을 사용하면 더 짧고 깔끔하게 표현할 수 있음



■ [코드 5-7]을 elif문을 활용하여 수정하기

[코드 5-8]

```
score = int(input("점수를 입력 ==> "))

if score >= 90 :
    print("A", end='')
elif score >= 80 :
    print("B", end='')
elif score >= 70 :
    print("C", end='')
elif score >= 60 :
    print("D", end='')
else :
    print("F", end='')
    print("학점입니다.")
```

[실행결과]

점수를 입력 ==> 77 ● 사용자 입력
C학점입니다.



확인문제

다음 지문에서 맞는 것에는 ○, 틀린 것에는 ×를 표시하십시오.

- ① if문 안에는 다른 if 구문이 들어갈 수 없다. ()
- ② 조건문에서 참일 때 수행하는 문장과 거짓일 때 수행하는 문장이 다르다면 if~else문을 사용한다. ()
- ③ 조건문에서 if문이 여러 번 중복해서 나오는 것을 중첩 if문이라 부른다. ()
- ④ 중첩 if문의 else~if를 줄이면 elseif로 사용할 수 있다. ()

정답

Click!



하나 더 알기 ✓

랜덤한 값 추출하기

랜덤(random)한 값이란 알 수 없는 임의의 숫자를 말합니다. 임의의 숫자를 뽑아야 하는 경우는 현실 세계에서도 종종 있습니다.

■ 로또 추천하기

로또는 1~45까지의 숫자를 뽑는 복권 게임입니다. 그런데 미리 어떤 숫자가 뽑힐지 프로그래밍해 놓는다면, 프로그램을 만든 사람이 항상 1등에 당첨될 것입니다. 그래서 로또를 뽑을 때마다 반드시 임의의 숫자가 나와야만 합니다. 파이썬에서는 랜덤한 값을 뽑기 위해 먼저 다음 코드가 필요합니다.

```
import random
```

이제부터는 랜덤한 숫자를 뽑을 수 있습니다. 1~45까지의 숫자를 추출하려면 다음과 같이 사용하면 됩니다.

```
random.randint(1,45)
```

`random.randint(시작, 끝)` 함수를 사용하면 시작~끝까지의 숫자 중, 절대로 예측할 수 없는 하나의 숫자가 랜덤하게 나옵니다. 이런 방식으로 공정하게 로또를 추천할 수 있습니다.

■ 점심 메뉴 고르기

`random.randint(시작, 끝)` 함수는 시작과 끝 사이의 숫자 중 임의로 한 가지를 추출합니다. 하지만 숫자가 아닌 경우는 어떻게 해야 할까요? 만약 점심메뉴를 김밥/라볶이/돈까스/짜장면 중에서 랜덤하게 고르고 싶다면 다음과 같이 사용하면 됩니다.

```
random.choice(["김밥", "라볶이", "돈까스", "짜장면"])
```

`random.choice([여러 개 값])` 함수는 여러 개의 값 중에서 한 가지를 랜덤하게 뽑아줍니다. 이때 랜덤하게 뽑힐 후보군은 꼭 `[]`로 묶어줘야 하는데, `[]`는 리스트를 의미합니다. 리스트는 7장에서 자세히 학습합니다.



PC방은 밤 10시가 넘으면 청소년은 출입할 수 없습니다.

밤 10시에 나이를 입력하고 출입 가능 여부를 체크하는 프로그램을 작성해 봅시다.



실행 결과

나이를 입력 ==> 17 ● — 사용자 입력

집에 갈 시간이네요!

협조 감사합니다.

1. lab05-01.py 파일을 만들고, 나이를 입력 받기

```
age = int(input("나이를 입력 ==> "))
```

2. 입력받은 나이가 18살 보다 어리다면 집으로 돌려보내도록 검사하는 조건문을 작성
 - 이때 청소년이든 아니든 협조 감사문을 출력하기

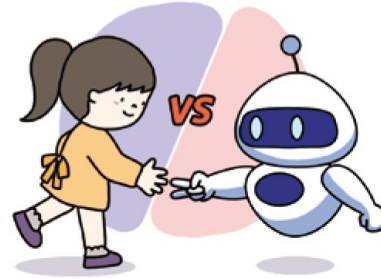
```
if age >= 18 :  
    print("즐거운 시간 되세요 ^^")  
else :  
    print("집에 갈 시간이네요!")  
  
print("협조 감사합니다.")
```

3. <Ctrl>+<S>를 눌러서 변경된 내용을 저장하고, <F5>를 눌러 실행 결과 확인하기

가위, 바위, 보 게임을 통해서 컴퓨터와 대결해 봅시다.

잘 알겠지만, 가위바위보 게임은 다음의 규칙을 따릅니다. 나를 기준으로 컴퓨터가 내는 것으로 확인하면 됩니다.

나	컴퓨터	결과(나 기준)
가위	가위 바위 보	비김 짐 이김
바위	가위 바위 보	이김 비김 짐
보	가위 바위 보	짐 이김 비김



실행 결과

나의 가위/바위/보 ==> 가위 ● — 사용자 입력
 컴퓨터의 가위/바위/보 ==> 보
 이겼습니다. ^^

실행 결과

나의 가위/바위/보 ==> 바위 ● — 사용자 입력
 컴퓨터의 가위/바위/보 ==> 보
 졌습니다. ππ

1. lab05-02.py 파일을 만들고, 컴퓨터가 랜덤하게 가위/바위/보를 내도록 random을 임포트하기

```
import random
```

2. 사용자의 가위/바위/보 중 하나를 입력받아 myHand 변수에 저장

```
myHand = input("나의 가위/바위/보 ==>")
```

3. 컴퓨터의 가위/바위/보는 랜덤하게 추출하여 comHand에 저장

```
comHand = random.choice(["가위", "바위", "보"])  
print("컴퓨터의 가위/바위/보 ==>", comHand)
```

4. 게임 진행하기 위해 사용자의 가위/바위/보에 따라 if문을 사용함
- 우선 사용자의 손이 '가위'일 때, 컴퓨터가 '가위'면 비김, 컴퓨터가 '바위'면 짐, 컴퓨터가 '보'면 이김이 되도록 코딩함
 - 규칙에 따라 사용자의 입장에서 이기고 지는 것을 판단함

```
if myHand == "가위" :  
    if comHand == "가위" :  
        print("비겼습니다. -.-")  
    elif comHand == "바위" :  
        print("졌습니다. πππ")  
    elif comHand == "보" :  
        print("이겼습니다. ^^")
```

5. 사용자의 손이 '바위'일 때와 '보'일 때도 위의 규칙에 맞게 작성함.
- 4번에서 가위일 때가 아닌 것으로 판단된 상태에서 '바위', '보'를 판단하는 것이므로 elif문을 사용

```
elif myHand == "바위" :  
    if comHand == "가위" :  
        print("이겼습니다. ^^")  
    elif comHand == "바위" :  
        print("비겼습니다. -.-")  
    elif comHand == "보" :  
        print("졌습니다. ㅠㅠ")  
  
elif myHand == "보" :  
    if comHand == "가위" :  
        print("졌습니다. ㅠㅠ")  
    elif comHand == "바위" :  
        print("이겼습니다. ^^")  
    elif comHand == "보" :  
        print("비겼습니다. -.-")
```


6. 사용자가 입력한 값이 가위/바위/보가 아닐 경우를 대비해 else문을 작성하기

```
else :  
    print("가위/바위/보 중 하나를 내세요.")
```

7. <Ctrl> + <S>를 눌러 저장한 후, <F5>를 눌러서 실행하고 결과를 확인함

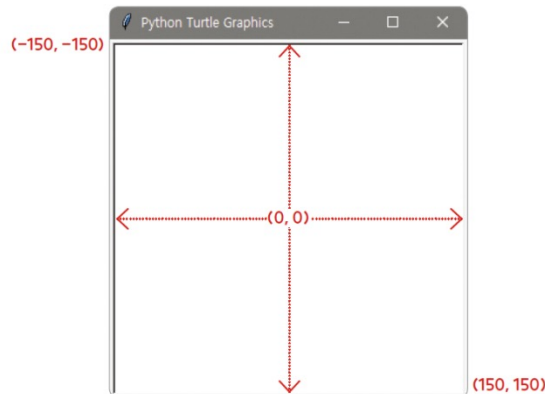
[실전 예제] 사춘기 거북이를 지도하기

실전 예제 사춘기 거북이를 지도하기

[문제]

청소년기의 거북이는 사춘기를 겪느라 언제 어디로 튈지 몰라서 그대로 둔다면 화면 밖으로 나가버리는 가출 거북이가 될 수도 있다. 그렇다고 해서 너무 제어하는 것도 좋지 않을 것 같아 청소년 거북이가 화면 안에서 노는 것은 그대로 두고, 화면 밖으로 나간다면 다시 화면의 중심으로 데려오자.

- 거북이가 활동할 화면의 크기는 (300, 300)으로 지정
- 화면의 중앙이 (0, 0)이므로 거북이의 활동 공간은 가로로 (-150~150), 세로로 (-150~150)임
- 거북이가 움직인 위치가 이 범위 안에 있으면 그대로 두고, 이 밖으로 나가면 중앙인 (0, 0) 위치로 데려다 놓기



실전 예제 사춘기 거북이를 지도하기

[해결]

```
import turtle
import random

turtle.shape("turtle")
turtle.pensize(5)
turtle.pencolor("blue")

turtle.screensize(300,300)
turtle.setup(330,330)

while True :
    angle = random.randint(0, 360)
    distance = random.randint(10, 100)
    turtle.right(angle)
    turtle.forward(distance)
    curX = turtle.xcor( )
    curY = turtle.ycor( )

    if (curX >= -150 and curX <= 150) and (curY >= -150 and curY <= 150) :
        print("Good Boy~")
    else :
        turtle.goto( 0, 0 )

turtle.done( )
```

감사합니다 :)