

# 불 자료형과 IF 조건문

# 목차

- 시작하기 전에
- else 조건문의 활용
- elif 구문
- if 조건문을 효율적으로 사용하기
- False로 변환되는 값
- pass 키워드
- 키워드로 정리하는 핵심 포인트
- 확인문제

**[핵심 키워드]** else 구문, elif 구문, False 값, pass

**[핵심 포인트]** if 조건문은 뒤에 else 구문을 붙여서 사용할 수 있다. 이처럼 if 구문 뒤에 else 구문을 붙인 것을 if else 조건문이라 부르기도 한다. 이것이 어떠한 경우에 사용하는 조건문인지 알아본다.

- 정반대되는 상황에서 두 번이나 if 조건문을 사용해 조건을 비교하는 것은 낭비일 수 있다.

```
01  # 입력을 받습니다.  
02  number = input("정수 입력> ")  
03  number = int(number)  
04  
05  # 짝수 조건  
06  if number % 2 == 0:  
07      print("짝수입니다")  
08  
09  # 홀수 조건  
10  if number % 2 == 1:  
11      print("홀수입니다")
```

- else 구문

- if 조건문 뒤에 사용하며, if 조건문의 조건이 거짓일 때 실행되는 부분

```
if 조건:
```

```
    조건이 참일 때 실행할 문장
```

```
else:
```

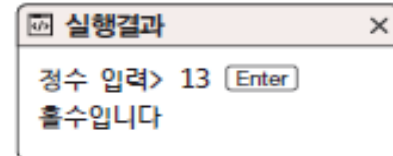
```
    조건이 거짓일 때 실행할 문장
```

□□□□는 들여쓰기 4칸

- 조건문이 오로지 두 가지로만 구분될 때 if else 구문을 사용하면 조건 비교를 단 한번만 하므로 이전의 코드보다 두 배 효율적

- 예시 - if 조건문에 else 구문 추가해서 짝수와 홀수 구분

```
01  # 입력을 받습니다.  
02  number = input("정수 입력> ")  
03  number = int(number)  
04  
05  # 조건문을 사용합니다.  
06  if number % 2 == 0:  
07      # 조건이 참일 때, 즉 짝수 조건  
08      print("짝수입니다")  
09  else:  
10      # 조건이 거짓일 때, 즉 홀수 조건  
11      print("홀수입니다")
```



- elif 구문

- 세 개 이상의 조건을 연결해서 사용
- if 조건문과 else 구문 사이에 입력

```
if 조건A:  
    조건A가 참일 때 실행할 문장  
elif 조건B:  
    조건B가 참일 때 실행할 문장  
elif 조건C:  
    조건C가 참일 때 실행할 문장  
...  
else:  
    모든 조건이 거짓일 때 문장
```

□□□□는 들여쓰기 4칸

## – 예시 – 계절 구하기

```
01  # 날짜/시간과 관련된 기능을 가져옵니다.
02  import datetime
03
04  # 현재 날짜/시간을 구하고
05  # 쉽게 사용할 수 있게 월을 변수에 저장합니다.
06  now = datetime.datetime.now()
07  month = now.month
08
09  # 조건문으로 계절을 확인합니다.
10  if 3 <= month <= 5:
11      print("현재는 봄입니다.")
12  elif 6 <= month <= 8:
13      print("현재는 여름입니다.")
14  elif 9 <= month <= 11:
15      print("현재는 가을입니다.")
16  else:
17      print("현재는 겨울입니다.")
```

실행결과

현재는 봄입니다



# if 조건문을 효율적으로 사용하기

## ● 조건문의 활용

### — 예시

조건	설명(학생 평가)	조건	설명(학생 평가)
4.5	신	1.75~2.3	오락문화의 선구자
4.2~4.5	교수님의 사랑	1.0~1.75	불가촉천민
3.5~4.2	현 체제의 수호자	0.5~1.0	자벌레
2.8~3.5	일반인	0~0.5	플랑크톤
2.3~2.8	일탈을 꿈꾸는 소시민	0	시대를 앞서가는 혁명의 씨앗

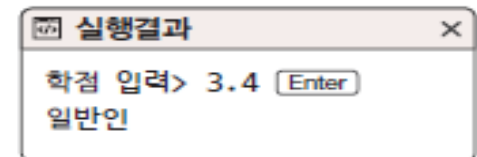
```

01  # 변수를 선언합니다.
02  score = float(input("학점 입력> "))
03
04  # 조건문을 적용합니다.
05  if score == 4.5:
06      print("신")
07  elif 4.2 <= score < 4.5:
08      print("교수님의 사랑")

```

# if 조건문을 효율적으로 사용하기

```
09 elif 3.5 <= score < 4.2:
10     print("현 체제의 수호자")
11 elif 2.8 <= score < 3.5:
12     print("일반인")
13 elif 2.3 <= score < 2.8:
14     print("일탈을 꿈꾸는 소시민")
15 elif 1.75 <= score < 2.3:
16     print("오락문화의 선구자")
17 elif 1.0 <= score < 1.75:
18     print("불가촉천민")
19 elif 0.5 <= score < 1.0:
20     print("자벌레")
21 elif 0 < score < 0.5:
22     print("플랑크톤")
23 elif score == 0:
24     print("시대를 앞서가는 혁명의 씨앗")
```



- 위에서 제외된 조건을 한 번 더 검사하여 비효율적

# if 조건문을 효율적으로 사용하기

```

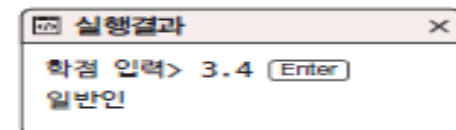
01  # 변수를 선언합니다.
02  score = float(input("학점 입력> "))
03
04  # 조건문을 적용합니다.
05  if score == 4.5:
06      print("신")
07  elif 4.2 <= score:
08      print("교수님의 사랑")
09  elif 3.5 <= score:
10      print("현 체제의 수호자")
11  elif 2.8 <= score:
12      print("일반인")
13  elif 2.3 <= score:
14      print("일탈을 꿈꾸는 소시민")
15  elif 1.75 <= score:
16      print("오락문화의 선구자")
17  elif 1.0 <= score:
18      print("불가촉천민")
19  elif 0.5 <= score:
20      print("자벌레")
21  elif 0 < score:
22      print("플랑크톤")
23  else:
24      print("시대를 앞서가는 혁명의 씨앗")
    
```

하위 값만 검사하고 상위 값은 검사를 생략

elif 4.2 <= score < 4.5:

→ elif 4.2 <= score:

조건 비교를 반으로 줄이고 코드 가독성 향상  
됨



# False로 변환되는 값

## ● 빈 컨테이너

- if 조건문의 매개변수에 볼 아닌 다른 값이 올 때 자동으로 볼로 변환
- 이 때 **False로 변환되는 값**: None, 0.0, 빈 문자열, 빈 바이트열, 빈 리스트

```
01 print("# if 조건문에 0 넣기")
02 if 0:
03     print("0은 True로 변환됩니다")
04 else:
05     print("0은 False로 변환됩니다")
06 print()
07
08 print("# if 조건문에 빈 문자열 넣기")
09 if "":
10     print("빈 문자열은 True로 변환됩니다")
11 else:
12     print("빈 문자열은 False로 변환됩니다")
```

실행결과

```
# if 조건문에 0 넣기
0은 False로 변환됩니다

# if 조건문에 빈 문자열 넣기
빈 문자열은 False로 변환됩니다
```

- 나중에 구현하고자 구문을 비워 두는 경우

```
if zero == 0
```

```
    빈 줄 삽입
```

```
else:
```

```
    빈 줄 삽입
```

---

```
01  # 입력을 받습니다.
```

```
02  number = input("정수 입력> ")
```

```
03  number = int(number)
```

```
04
```

```
05  # 조건문 사용
```

```
06  if number > 0:
```

```
07      # 양수일 때: 아직 미구현 상태입니다.
```

```
08  else:
```

```
09      # 음수일 때: 아직 미구현 상태입니다.
```

---

- IndentationError

- if 조건문 사이에는 무조건 들여쓰기 4칸 넣고 코드 작성해야 함

- pass 키워드

- 아무것도 작성하지 않고 임시적으로 비워 둠

---

```
01  # 입력을 받습니다.
02  number = input("정수 입력> ")
03  number = int(number)
04
05  # 조건문 사용
06  if number > 0:
07      # 양수일 때: 아직 미구현 상태입니다.
08      pass
09  else:
10      # 음수일 때: 아직 미구현 상태입니다.
11      pass
```

---

- **else 구문** : if 조건문 뒤에 사용하며, if 조건문의 조건이 거짓일 때 실행
- **elif 구문** : if 조건문과 else 구문 사이에 입력하며, 세 개 이상의 조건을 연결해서 사용할 때 적절
- **False로 변환되는 값** : if 조건문의 조건식에서 False로 변환되는 값은 None, 0, 0.0, 빈 문자열, 빈 바이트 열, 빈 리스트, 빈 튜플, 빈 딕셔너리 등이 있음
- **pass 키워드** : 프로그래밍의 전체 골격을 잡아두고 내부에 처리할 내용은 나중에 만들고자 할 때 pass 키워드 입력

- 다음 코드의 실행결과를 예측해 빈칸에 결괏값을 입력하세요. 아래의 코드는 모두 같고 입력 결과가 다른 경우입니다.

```
x = 2  
y = 10
```

```
if x > 4:  
    if y > 2:  
        print(x * y)  
else:  
    print(x + y)
```

```
x = 1  
y = 4
```

```
if x > 4:  
    if y > 2:  
        print(x * y)  
else:  
    print(x + y)
```

```
x = 10  
y = 2
```

```
if x > 4:  
    if y > 2:  
        print(x * y)  
else:  
    print(x + y)
```



- 다음 중첩 조건문에 논리 연산자 적용해 하나의 if 조건문으로 만들어 주세요.

```
if x > 10:  
    if x < 20:  
        print("조건에 맞습니다.")
```



```
print("조건에 맞습니다.")
```