

# 불 자료형과 IF 조건문

# 목차

- 시작하기 전에
- 불 만들기 : 비교 연산자
- 불 연산하기 : 논리 연산자
- 논리 연산자의 활용
- if 조건문이란
- 날짜/시간 활용하기
- 컴퓨터의 조건
- 키워드로 정리하는 핵심 포인트
- 확인문제

**[핵심 키워드]** 불, 비교 연산자, 논리 연산자, if 조건문

**[핵심 포인트]** 프로그래밍 언어에는 기본적인 자료형으로 참과 거짓을 나타내는 값이 있으며, 이를 불(boolean)이라 한다. 불 자료를 만드는 방법과 이에 관련된 연산자에 대해 알아본다.

- Boolean
  - 불린 / 불리언 / 불
  - True와 False 값만 가질 수 있음

```
>>> print(True)
True
>>> print(False)
False
```

| 연산자 | 설명  | 연산자 | 설명     |
|-----|-----|-----|--------|
| ==  | 같다  | >   | 크다     |
| !=  | 다르다 | <=  | 작거나 같다 |
| <   | 작다  | >=  | 크거나 같다 |

- 숫자 또는 문자열에 적용

```
>>> print(10 == 100)
False
>>> print(10 != 100)
True
>>> print(10 < 100)
True
>>> print(10 > 100)
False
>>> print(10 <= 100)
True
>>> print(10 >= 100)
False
```

| 조건식       | 의미               | 결과 |
|-----------|------------------|----|
| 10 == 100 | 10과 100은 같다      | 거짓 |
| 10 != 100 | 10과 100은 다르다     | 참  |
| 10 < 100  | 10은 100보다 작다     | 참  |
| 10 > 100  | 10은 100보다 크다     | 거짓 |
| 10 <= 100 | 10은 100보다 작거나 같다 | 참  |
| 10 >= 100 | 10은 100보다 크거나 같다 | 거짓 |

- 문자열에도 비교 연산자 적용 가능

```
>>> print("가방" == "가방")
True
>>> print("가방" != "하마")
True
>>> print("가방" < "하마")
True
>>> print("가방" > "하마")
False
```

# 불 연산하기 : 논리 연산자

- 불끼리 논리 연산자 사용 가능

| 연산자 | 의미  | 설명  |
|-----|-----|---|
| not | 아니다 | 불을 반대로 전환합니다.   |
| and | 그리고 | 피연산자 두 개가 모두 참일 때 True를 출력하며, 그 외는 모두 False를 출력합니다.           |
| or  | 또는  | 피연산자 두 개 중에 하나만 참이라도 True를 출력하며, 두 개가 모두 거짓일 때만 False를 출력합니다. |

- not 연산자

- 단항 연산자
- 참과 거짓 반대로 바꿈

```
>>> print(not True)
False
>>> print(not False)
True
```

## – 예시 – not 연산자 조합하기

```
01 x = 10
02 under_20 = x < 20
03 print("under_20:", under_20)
04 print("not under_20:", not under_20)
```

실행결과

under\_20: True  
not under\_20: False



- and 연산자와 or 연산자

- and 연산자는 양쪽 변의 값이 모두 참일 때만 True를 결과로 냄

- and 연산자

| 좌변    | 우변    | 결과    |
|-------|-------|-------|
| True  | True  | True  |
| True  | False | False |
| False | True  | False |
| False | False | False |

- or 연산자

| 좌변    | 우변    | 결과    |
|-------|-------|-------|
| True  | True  | True  |
| True  | False | True  |
| False | True  | True  |
| False | False | False |

## – 예시 – and 연산자와 or 연산자

"사과 그리고 배 가져와!"

"사과 또는 배 가져와!"

"치킨(True) 그리고 쓰레기(False) 가져와!"

"치킨(True) 또는 쓰레기(False) 가져와!"

```
>>> print(True and True)
True
>>> print(True and False)
False
>>> print(False and True)
False
>>> print(False and False)
False
>>> print(True or True)
True
>>> print(True or False)
True
>>> print(False or True)
True
>>> print(False or False)
False
```

- and 연산자



- or 연산자



- if 조건문

- 조건에 따라 코드 실행하거나 실행하지 않게 할 때 사용하는 구문
- 조건 분기

if 불 값이 나오는 표현식: → if의 조건문 뒤에는 반드시 콜론(:)을 붙여줘야 합니다.

□□□ 불 값이 참일 때 실행할 문장

□□□ 불 값이 참일 때 실행할 문장

□□□□는 들여쓰기 4칸

↓  
if문 다음 문장은 4칸 들여쓰기 후 입력합니다.

## — 예시

```
>>> if True: Enter
    print("True입니다...!") Enter
    print("정말 True입니다...!") Enter
    Enter
True입니다...!
정말 True입니다...!
```

```
>>> if False: Enter
    print("False입니다...!") Enter
    Enter
>>>
```

## 예시 - 조건문의 기본 사용

```
01  # 입력을 받습니다.  
02  number = input("정수 입력> ")  
03  number = int(number)  
04  
05  # 양수 조건  
06  if number > 0:  
07      print("양수입니다")  
08  
09  # 음수 조건  
10  if number < 0:  
11      print("음수입니다")  
12  
13  # 0 조건  
14  if number == 0:  
15      print("0입니다")
```

**실행결과 1** ×

정수 입력> 273   
양수입니다

**실행결과 2** ×

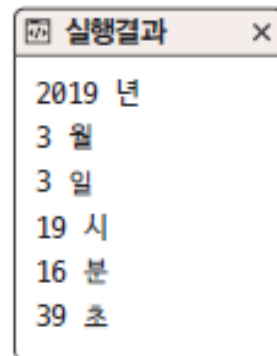
정수 입력> -52   
음수입니다

**실행결과 3** ×

정수 입력> 0   
0입니다

- 예시 - 날짜/시간 출력하기
  - datetime.datetime.now() 함수

```
01  # 날짜/시간과 관련된 기능을 가져옵니다.
02  import datetime
03
04  # 현재 날짜/시간을 구합니다.
05  now = datetime.datetime.now()
06
07  # 출력합니다.
08  print(now.year, "년")
09  print(now.month, "월")
10  print(now.day, "일")
11  print(now.hour, "시")
12  print(now.minute, "분")
13  print(now.second, "초")
```



실행결과

2019 년  
3 월  
3 일  
19 시  
16 분  
39 초

## 예시 - 날짜/시간을 한 줄로 출력하기

```
01  # 날짜/시간과 관련된 기능을 가져옵니다.  
02  import datetime  
03  
04  # 현재 날짜/시간을 구합니다.  
05  now = datetime.datetime.now()  
06  
07  # 출력합니다.  
08  print("{}년 {}월 {}일 {}시 {}분 {}초".format(  
09      now.year,  
10      now.month,  
11      now.day,  
12      now.hour,  
13      now.minute,  
14      now.second  
15  ))
```

실행결과

2019년 3월 3일 19시 18분 45초



## 예시 - 오전과 오후를 구분하는 프로그램

```
01  # 날짜/시간과 관련된 기능을 가져옵니다.  
02  import datetime  
03  
04  # 현재 날짜/시간을 구합니다.  
05  now = datetime.datetime.now()  
06  
07  # 오전 구분  
08  if now.hour < 12:  
09      print("현재 시각은 { }시로 오전입니다!".format(now.hour))  
10  
11  # 오후 구분  
12  if now.hour >= 12:  
13      print("현재 시각은 { }시로 오후입니다!".format(now.hour))
```

실행결과

현재 시각은 19시로 오후입니다!

## – 예시 – 계절을 구분하는 프로그램

```
01  # 날짜/시간과 관련된 기능을 가져옵니다.  
02  import datetime  
03  
04  # 현재 날짜/시간을 구합니다.  
05  now = datetime.datetime.now()  
06  
07  # 봄 구분  
08  if 3 <= now.month <= 5:  
09      print("이번 달은 {}월로 봄입니다!".format(now.month))  
10  
11  # 여름 구분  
12  if 6 <= now.month <= 8:
```

```
13     print("이번 달은 {}월로 여름입니다!".format(now.month))
14
15     # 가을 구분
16     if 9 <= now.month <= 11:
17         print("이번 달은 {}월로 가을입니다!".format(now.month))
18
19     # 겨울 구분
20     if now.month == 12 or 1 <= now.month <= 2:
21         print("이번 달은 {}월로 겨울입니다!".format(now.month))
```

실행결과

이번 달은 3월로 봄입니다!

- if 조건문의 형식

if 불 값이 나오는 표현식:

□□□□ 불 값이 참일 때 실행할 문장

□□□□는 들여쓰기 4칸

```
01  # 입력을 받습니다.
02  number = input("정수 입력> ")
03
04  # 마지막 자리 숫자를 추출
05  last_character = number[-1]
06
07  # 숫자로 변환하기
08  last_number = int(last_character)
09
```

```
10  # 짝수 확인
11  if last_number == 0 \
12      or last_number == 2 \
13      or last_number == 4 \
14      or last_number == 6 \
15      or last_number == 8:
16      print("짝수입니다")
17
18  # 홀수 확인
19  if last_number == 1 \
20      or last_number == 3 \
21      or last_number == 5 \
22      or last_number == 7 \
23      or last_number == 9:
24      print("홀수입니다")
```

실행결과 1

정수 입력> 52

짝수입니다

실행결과 2

정수 입력> 273

홀수입니다

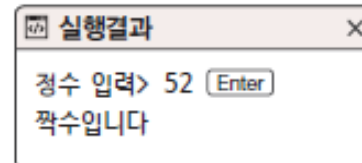
## 예시 - in 연산자를 활용한 수정

```
01  # 입력을 받습니다.  
02  number = input("정수 입력> ")  
03  last_character = number[-1]  
04  
05  # 짝수 조건  
06  if last_character in "02468":  
07      print("짝수입니다")  
08  
09  # 홀수 조건  
10  if last_character in "13579":  
11      print("홀수입니다")
```



## 예시 - 나머지 연산자를 활용한 짝수와 홀수 구분

```
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")
```



- **불** : 파이썬의 기본 자료형으로 True와 False 나타내는 값
- **비교 연산자** : 숫자 또는 문자열에 적용하며 대소 비교하는 연산자
- **논리 연산자** : not, and, or 연산자 있으며, 불 만들 때 사용
- **if 조건문** : 조건에 따라 코드 실행하거나 실행하지 않게 만들고 싶을 때 사용



- 비교 연산자를 사용한 조건식입니다. 결과가 참이면 True를, 거짓이면 False를 적어 보세요.

| 조건식       | 결과                   |
|-----------|----------------------|
| 10 == 100 | <input type="text"/> |
| 10 != 100 | <input type="text"/> |
| 10 > 100  | <input type="text"/> |
| 10 < 100  | <input type="text"/> |
| 10 <= 100 | <input type="text"/> |
| 10 >= 100 | <input type="text"/> |

- 몇 번인가요?

```
x = 2
if x > 4:
    print("참입니다")
```

①

```
x = 1
if x > 4:
    print("참입니다")
```

②

```
x = 10
if x > 4:
    print("참입니다")
```

③

- 사용자로부터 숫자 두 개를 입력받고 첫 번째 입력받은 숫자가 큰지, 두 번째 입력받은 숫자가 큰지를 구하는 프로그램을 다음 빈칸을 채워 완성해 보세요.

```
a =  (input("> 1번째 숫자: "))
b =  (input("> 2번째 숫자: "))
print()

if  :
    print("처음 입력했던 {}가 {}보다 더 큼니다".)
if  :
    print("두 번째로 입력했던 {}가 {}보다 더 큼니다".)
```

## 실행결과

```
> 1번째 숫자: 100 
> 2번째 숫자: 10 
```

처음 입력했던 100가 10보다 더 큼니다