2주차

2-1. 불 자료형

Boolean

- 불린, 불리언 등으로 부름
- 짧게 `Bool` 이라고 하기도 하고, 불이라고 자주 말 함
    - 오직 `True` (참), `False` (거짓)의 두 값만 가질 수 있음

```
# 문자열이 아니다!
print(True)
print(False)
```

불 만들기: 비교 연산자

연산자	설명
==	같다
!=	다르다
<	작다
>	크다
<=	작거나 같다(이하)
>=	크거나 같다(이상)

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

```
# 문자열에서의 비교 연산자
print("가방" == "가방")
print("가방" != "가방")
```

```
print("가방" < "하마")
print("가방" > "하마")
```

```
# 범위 구하기
x = 25
print(10 < x < 30)
print(40 < x < 60)
```

불 만들기: 논리 연산자

연산자	의미	설명
not	아니다	불 값을 반대로 전환
and	그리고	피연산자 두 개가 모두 참 일때 True, 그 외는 모두 False
or	또는	피연산자 두 개 중에 하나만 참이라도 True, 그 외 모두 거짓이면 False

```
# not 실습
print(not True)
print(not False)
```

```
# and 실습
print(True and True)
print(True and False)
print(False and True)
print(False and False)
```

```
# or실습
print(True or True)
print(True or False)
print(False or True)
print(False or False)
```

2-2. `if` 조건문

if 조건문이란?

- 조건에 따라 코드를 실행하거나 실행하지 않게 만들고 싶을 때 사용
- 코드의 실행 흐름을 변경
- '조건 분기'

```
if True:
    print("True입니다.")

if False:
    print("False입니다.")
```

```
# if문 기본 사용
# 입력을 받습니다.
numebr = int(input())

# 양수 조건
if number > 0:
    print('양수입니다.')

# 음수 조건
if number < 0:
    print('음수입니다.')

# 0 조건
if number == 0
    print('0입니다.')
```

날짜/시간을 활용하기

```
# 날짜/시간과 관련된 기능을 가져옵니다.
import datetime

# 현재 날짜/시간을 구합니다.
now = datetime.datetime.now()

# 출력합니다.
```

```
print(now.year, "년")
print(now.month, "월")
print(now.day, "일")
print(now.hour, "시")
print(now.minute, "분")
print(now.second, "초")
```

- 오전과 오후를 구분하는 프로그램

```
# 날짜/시간과 관련된 기능을 가져옵니다.
import datetime

# 현재 날짜/시간을 구합니다.
now = datetime.datetime.now()

# 오전 구분
if now.hour < 12:
    print("현재 시각은", now.hour, "시로 오전입니다!")

# 오후 구분
if now.hour >= 12:
    print("현재 시각은", now.hour, "시로 오후입니다!")
```

- 계절을 구분하는 프로그램

```
# 날짜/시간과 관련된 기능을 가져옵니다.
import datetime

# 현재 날짜/시간을 구합니다.
now = datetime.datetime.now()

# 봄 구분
if 3 <= now.month <= 5:
    print("이번 달은", now.month, "월로 봄입니다!")

# 여름 구분
if 6 <= now.month <= 8:
    print("이번 달은", now.month, "월로 여름입니다!")
```

```
# 가을 구분
if 9 <= now.month <= 11:
    print("이번 달은", now.month, "월로 가을입니다!")

# 겨울 구분
if now.month == 12 or 1 <= now.month <= 2:
    print("이번 달은", now.month, "월로 겨울입니다!")
```

컴퓨터의 조건(짝수와 홀수 구분)

- 끝자리 숫자로 확인

```
# 입력을 받습니다.
number = input()

# 마지막 자리 숫자를 추출
last_character = number[-1]

# 숫자로 변환하기
last_number = int(last_character)

# 짝수 확인
if last_number == 0 \
    or last_number == 2 \
    or last_number == 4 \
    or last_number == 6 \
    or last_number == 8:
    print("짝수입니다.")

# 홀수 확인
if last_number == 1 \
    or last_number == 3 \
    or last_number == 5 \
    or last_number == 7 \
    or last_number == 9:
```

```
    print("홀수입니다.")
```

- `in` 문자열 연산자를 활용하여 확인

```
# 입력을 받습니다.
number = input()

# 마지막 자리 숫자를 추출
last_character = number[-1]

# 짝수 조건
if last_character in "02468":
    print("짝수입니다.")

# 홀수 조건
if last_character in "13579":
    print("홀수입니다.")
```

- 나머지 연산자를 활용하여 확인

```
# 입력을 받습니다.
number = input()

# 숫자 정수 형식으로 변환
number = int(number)

# 짝수 조건
if number % 2 == 0:
    print("짝수입니다.")

# 홀수 조건
if number % 2 == 1:
    print("홀수입니다.")
```

2-3. `if` ~ `else` 와 `elif` 구문

- 두 가지로만 구분 될 때가 많음(오전/오후 등)

```
# 입력을 받습니다.
number = int(input())

# 짝수 조건
if number % 2 == 0:
    print("짝수입니다.")

# 홀수 조건
if number % 2 == 1:
    print("홀수입니다.")
```

else 조건문

```
# 입력을 받습니다.
number = int(input())

# 짝수 조건
if number % 2 == 0:
    print("짝수입니다.")
else:
    print("홀수입니다.")
```

- 본래 두 번의 조건문을 사용하여 비교하였으나 `if else` 를 사용하여 조건 비교를 한 번만 하므로 이전의 코드보다 효율적임

elif 구문

- 다양한 조건을 연결

```
# 날짜/시간과 관련된 기능을 가져옵니다.
import datetime

# 현재 날짜/시간을 구합니다.
now = datetime.datetime.now()

# 조건문으로 계절 구분
if 3 <= now.month <= 5:
    print("이번 달은", now.month, "월로 봄입니다!")
```

```
elif 6 <= now.month <= 8:
    print("이번 달은", now.month, "월로 여름입니다!")
elif 9 <= now.month <= 11:
    print("이번 달은", now.month, "월로 가을입니다!")
else:
    print("이번 달은", now.month, "월로 겨울입니다!")
```

if 조건문을 효율적으로 사용하기

조건	설명
4.0~4.5	A
3.5~4.0	B
3.0~3.5	C
2.5~3.0	D
~2.5	F

```
# 입력을 받습니다.
score = float(input())

# 조건문을 적용
if 4.0 <= score <= 4.5:
    print("A")
elif 3.5 <= score < 4.0:
    print("B")
elif 3.0 <= score < 3.5:
    print("C")
elif 2.5 <= score < 3.0:
    print("D")
else:
    print("F")
```

- 코드를 더 효율적으로

```
# 입력을 받습니다.
score = float(input())

# 조건문을 적용
```

```
if 4.0 <= score <= 4.5:
    print("A")
elif 3.5 <= score:
    print("B")
elif 3.0 <= score:
    print("C")
elif 2.5 <= score:
    print("D")
else:
    print("F")
```

False로 변환되는 값

```
print('# if 조건문에 0 넣기')
if 0:
    print("0은 True로 변환됩니다.")
else:
    print("0은 False로 변환됩니다.")
print()

print("# if 조건문에 빈 문자열 넣기")
if "":
    print("빈 문자열은 True로 변환됩니다.")
else:
    print("빈 문자열은 False로 변환됩니다.")
```

`pass` 키워드

- 조건문 내의 부분을 실행하지 않고 넘어갈 때 사용

```
number = int(input())

if(number >= 0):
    pass
else:
    print(number, "은 음수입니다.")
```


PROJECT:
DAWN
start