

파이썬프로그래밍 2분반 ## <6주차 과제> ---

정보보안공학과 ### 202121556 ### 곽지현

2023-04-13

In [1]: # 1. 비교 연산자의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.
 # Boolean, 비교 연산자

```
print(True == True) # 'True와 True는 같다' 참이므로 True를 출력
print(True < True) # 'True는 True미만이다' 거짓으로 False를 출력
print(True > True) # 'True는 True초과이다' 거짓으로 False를 출력
print(True <= True) # 'True는 True보다 작거나 같다' 참이므로 True를 출력
print(True >= True) # 'True는 True보다 크거나 같다' 참이므로 True를 출력

print(True == False) # 'True와 False는 같다' 거짓으로 False를 출력
print(True != False) # 'True와 False는 같지 않다' 참이므로 True를 출력
```

True
 False
 False
 True
 True
 False
 True

In [2]: # 2. 논리 연산자(not)의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.
 # Boolean, 비교 연산자, 논리 연산자

```
print(not True) # True의 반대인 False를 출력
print(not False) # False의 반대인 True를 출력
print((not True) == (not False)) # 'True의 반대인 False와 True의 반대인 False가 같다'
```

False
 True
 True

In [3]: # 3. 논리 연산자(and, or)의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.
 # Boolean, 논리 연산자

```
print(True and True and True)
# 세 개 다 참이므로 True를 출력
print(True and False and True)
# 두 개가 참이고 한 개가 거짓으로 False를 출력
print(True and (False or True))
# 한 개가 참이고 한 개가 거짓 혹은 참이므로 True를 출력
print(True and (False and True))
# 한 개가 참이고 한 개가 거짓 그리고 참이므로 False를 출력
print(True or (False and True))
# 참 이거나 (거짓 그리고 참) 이므로 True를 출력
```

```
True
False
True
False
True
```

In [4]: # 4. if문 형식의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.
Boolean, 논리 연산자, 인덴트(Indent)

```
if True: print("if문의 조건문의 결과가 True일 때 실행됨")
    # if문의 조건문 결과가 참이므로 실행됨
if False: print("False일 때는 실행 안 됨")
    # if문의 조건문 결과가 거짓이어서 실행 안 됨

if not False:
    print("논리 연산(not)의 결과가 True일 때 실행됨")
    # 거짓의 반대인 참의 결과로 실행됨
if True & True:
    print("논리 연산(and)의 결과가 True일 때 실행됨")
    # 참 그리고 참 이므로 실행됨
if True | True:
    print("논리 연산(or)의 결과가 True일 때 실행됨")
    # 참 혹은 참 이므로 실행됨
if True and True:
    print("논리 연산(and)의 결과가 True일 때 실행됨")
    # 참 그리고 참 이므로 실행됨
if True or True:
    print("논리 연산(or)의 결과가 True일 때 실행됨")
    # 참 혹은 참 이므로 실행됨
```

```
if문의 조건문의 결과가 True일 때 실행됨
논리 연산(not)의 결과가 True일 때 실행됨
논리 연산(and)의 결과가 True일 때 실행됨
논리 연산(or)의 결과가 True일 때 실행됨
논리 연산(and)의 결과가 True일 때 실행됨
논리 연산(or)의 결과가 True일 때 실행됨
```

In [1]: # 5. in if . 연산자를 이용한 문의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.
Boolean, 논리 연산자, in 연산자, if문 elif문, format 함수, int 함수, 문자열 슬라이싱,

```
num = input("정수 입력>") # 정수를 입력 받아 num의 변수에 대입
last_character = num[-1] # last_character 변수에 num-1의 값을 대입

if last_character in "02468": # last_character의 값이 "02468"안에 있다면
    print("짝수입니다.") # "짝수입니다."를 출력
else:
    print("홀수입니다.") # "02468"안에 없다면 "홀수입니다."를 출력

if int(num) in [0,1,2,3,4,5,6,7,8,9]:
    # 입력받은 정수가 "0,1,2,3,4,5,6,7,8,9"안에 있다면
    print("입력된 {}은 10미만의 수입니다.".format(num))
    # "입력된 {}은 10미만의 수입니다."와 num의 값을 출력
elif int(num) >= 10:
    # 입력받은 정수가 10이상이라면
    print("입력된 {}은 10이상의 수입니다.".format(num))
    # "입력된 {}은 10이상의 수입니다."와 num의 값을 출력
```

```
정수 입력>23
홀수입니다.
입력된 23은 10이상의 수입니다.
```

In [2]: # 6. elif문의 이해를 위해 다음 코드를 실행하여 결과를 확인해보세요.
if문, elif문, 비교 연산자, 자료형 캐스팅(int 함수), input 함수

```
num = int(input("0~100 정수 입력 >")) # 숫자를 입력 받아 int로 캐스팅

if (num >= 100): # num의 값이 100이상이라면
    print("1등급") # "1등급"을 출력
elif(num >= 90): # num의 값이 90이상이라면
    print("2등급") # "2등급"을 출력
elif(num >= 80): # num의 값이 80이상이라면
    print("3등급") # "3등급"을 출력
elif(num >= 70): # num의 값이 70이상이라면
    print("4등급") # "4등급"을 출력
elif(num >= 60): # num의 값이 60이상이라면
    print("5등급") # "5등급"을 출력
elif(num >= 50): # num의 값이 50이상이라면
    print("6등급") # "6등급"을 출력
elif(num >= 40): # num의 값이 40이상이라면
    print("7등급") # "7등급"을 출력
elif(num >= 30): # num의 값이 30이상이라면
    print("8등급") # "8등급"을 출력
else: # 다른 값이면
    print("X") # "X"를 출력
```

0~100 정수 입력 >78

4등급

In [10]: # 13. 숫자 2개와 연산자 1개를 입력받아 연산자에 따라 계산해주는 프로그램입니다.
다음과 같은 결과가 나오도록 빈칸을 채워 실행하세요.

```
# input 함수, 비교 연산자, Boolean, 자료형 캐스팅(?), 문자열 형식화
# if문

num1 = input("정수 입력 > ") # 숫자를 입력 받아 num1에 대입
num2 = input("정수 입력 > ") # 숫자를 입력 받아 num2에 대입
op = input("+, -, *, / 중 하나 입력 > ") # "+, -, *, /"중 입력 받아 op에 대입
result = 0 # result 변수에 0을 대입

if op == "+": # op의 값이 "+"와 같으면
    result = int(num1) + int(num2) # result에 num1과 num2를 더한 값을 대입
if op == "-": # op의 값이 "-"과 같으면
    result = int(num1) - int(num2) # result에 num1에서 num2를 뺀 값을 대입
if op == "*": # op의 값이 "*"와 같으면
    result = int(num1) * int(num2) # result에 num1과 num2를 곱한 값을 대입
if op == "/": # op의 값이 "/"와 같으면
    result = int(num1) / int(num2) # result에 num1 나누기 num2를 한 값을 대입

print("{} {} {} = {}".format(num1, op, num2, result)) # num1, op, num2, result 값을 출력
```

정수 입력 > 1

정수 입력 > 2

+, -, *, / 중 하나 입력 > +

1 + 2 = 3

In [17]: # 14. 13번 프로그램에서 연산자 입력의 예외처리가 추가된 프로그램입니다.
다음과 같은 결과가 나오도록 빈칸을 채워 실행하세요.

```
num1 = input("정수 입력 > ") # 숫자를 입력 받아 num1에 대입
num2 = input("정수 입력 > ") # 숫자를 입력 받아 num2에 대입
op = input("+, -, *, / 중 하나 입력 > ") # "+, -, *, /"중 입력 받아 op에 대입
```

```
result = 0 # result 변수에 0을 대입

# 논리 연산자를 사용하지 않고, +, -, *, /일 때만 계산되도록 조건식 작성
if op in "+-*":"/": # op의 값이 "+-*/"중에 있다면
    if op == "+": # op의 값이 "+"와 같으면
        result = int(num1) + int(num2) # result에 num1과 num2를 더한 값을 대입
    if op == "-": # op의 값이 "-"와 같으면
        result = int(num1) - int(num2) # result에 num1에서 num2를 뺀 값을 대입
    if op == "*": # op의 값이 "*"와 같으면
        result = int(num1) * int(num2) # result에 num1과 num2를 곱한 값을 대입
    if op == "/": # op의 값이 "/"와 같으면
        result = int(num1) / int(num2) # result에 num1 나누기 num2를 한 값을 대입

print("{} {} {} = {}".format(num1,op,num2,result))
# num1,op,num2,result 값을 출력

else: # op의 값이 "+-*/"중에 없다면
    print("+, -, *, / 이외의 문자를 입력했습니다.")
    # "+, -, *, / 이외의 문자를 입력했습니다."를 출력
```

정수 입력 > 1
정수 입력 > 2
+, -, *, / 중 하나 입력 > !
+, -, *, / 이외의 문자를 입력했습니다.