05.Python 연산자

1. Python 연산자

1. 산술 연산자
   - 산술 연산자: 더하기, 빼기, 곱하기, 나누기와 같은 연산자
   - 단항 연산자: 피 연산자가 두 개가 아닌 하나인 연산자
   - 단항 연산자는 가장 우선순위가 높아서 먼저 수행
   - 지수 연산자: **
   - 결합순서: 오른쪽 → 왼쪽
   - %: 나머지를 구해주는 연산자
   - 나머지가 나올 수 있는 수: 0과 1

연산자	우선순위	설명	결합순서
+, -	1	단항 연산자	-
**	2	지수 연산자	왼쪽 <- 오른쪽
*,/,%,//	3	곱하기,나누기,나머지,몫	왼쪽 -> 오른쪽
+, -	4	더하기,빼기	왼쪽 -> 오른쪽

2] 적용 예시

```
print  ("  2 ** 2 ** 3" , 2 ** 2 ** 3)    # ** 연산자의 결합순서는 오른쪽에서 왼쪽
print ("3 + 5.0" , 3 + 5.0)  # 정수 + 실수의 결과는 실수

print ("5 / 2.0 " , 5 / 2.0) # 정수 / 실수의 결과는 실수
print ("5 / 2" , 5 / 2)
a = 5 / 3
b = 5 % 3
print ("a, b " , a, b)
print ("divmod(5,3) " , divmod(5,3))


print ("5 / 3 " ,5 / 3)   # 1.6666...7
print ("5 //  " ,5 // 3)  # 1

print ("-7/4 " ,-7/4)   # -7을 4로 나눈다 ,  -1.75
print ("-7//4 " ,-7//4) # -7을 4로 나눈다 , -1.75 보다 더 작은 정수를 찾는 것 = -2
-
```

1. Python 연산자

2. 관계연산자

- 객체가 지닌 값의 크기(대소)를 비교하여 True 또는 False를 반환
- ==: 두 개의 객체가 동일한지를 판단하는 연산자
- 양쪽에 있는 두 객체의 값이 동일해야 ture
- !=: 두 개의 객체 값이 달라야 true

2] 적용 예시

```
print   ("  6 == 9" , 6 == 9)    # False
print   ("  6 != 9" , 6 != 9)      # True
print   ("   1 > 3" , 1 > 3)
print   ("  4 <= 5" , 4 <= 5)
a = 5
b = 10
print   ("  a < b" , a < b)

print ("--------------------------")
x = [1,2,3]
y = [1,2,3]
z = y
print  (" x ->" , x)
print  (" y ->" , y)
print  (" z ->" , z)
print  (" x == y" , x == y)
print  (" x == z" , x == z)
print  (" x is y" , x is y)  # is: x의 식별자와 y의 식별자를 비교 = 같은 객체인지 비교
print  ("id(x) ", id(x) )
print  ("id(y) ", id(y) )
print  (" x is z" , x is z)   # x is z False
print  (" y is z" , y is z)   # y is z True
```

1. Python 연산자

3. **논리 연산자**

- 피연산자의 값으로 진리값인 True 또는 False을 취하여 논리 적인 계산을 수행하는 연산자
- and , or , not

2] 적용 예시 1

```
a = 20
b = 30
print  (" a > 10 and b < 50 " , a > 10 and b < 50)   # True
print  ("--------------------------")
print  ("True + 1 "   ,True + 1 )   # true가 1이니까 1을 출력
print  ("False + 1 "  ,False + 1)
print  ("False * 75 " ,False * 75)  # false가 0 → 0*75 = 0이 출력
print  ("True * 75 "  ,True * 75)
print  ("--------------------------")
print  ("bool(0) "     , bool(0)) # 정수 0은 거짓
print  ("bool(1) "     , bool(1))
print  ("bool(100) "   , bool(100))
print  ("bool(-100) "  , bool(-100))
print
print  ("bool(0.0) "   , bool(0.0)) # 실수 0.0은 거짓
print  ("bool(0.1) "   , bool(0.1)) # 0.0이 아닌 나머지 모든 실수 = true
print  ("--------------------------")
print  ("bool('abc') "     , bool('abc')) # 일반적인 문자열(ex. abc) → true
print  ( "bool('') "     , bool(''))        # 공백조차 없는 비어있는 문자열 → false
print
print  ("boolbool([]) "     , bool([])) # 공 리스트는 거짓
print  ("boolbool([1,2,3]) "     , bool([1,2,3]))
print
print  ("bool(()) "     , bool(())) # 공 튜플은 거짓
print  ("bool((1,2,3)) "     , bool((1,2,3)))
print
print  ("bool({}) "     , bool({})) # 공 사전은 거짓
print  ("bool({1:2}) "     , bool({1:2}))
print
```

2] 적용 예시 2

```
print ("--------------------------")
print ("1 and 0 "     , 1 and 0)
print ("0 or 0 "      , 0 or 0)
print ("1 or 0 "      , 1 or 0)
print ()
print ("[] or 1 "     , [] or 1) # [] 거짓
print ("[] or () "    , [] or ()) # [], () 거짓
print ("[] and 1  "   , [] and 1) # [] 거짓이므로 1은 참조할 필요 없
print ("--------------------------")
print ("1 and 2 "     , 1 and 2)
print ("1 or 2  "     , 1 or 2)
```