

Day 5 - Python Operator

파이썬에서 불린(Boolean)으로 평가되는 주요 값

1. True로 평가되는 주요 값

- a. 0이 아닌 모든 숫자 (양수, 음수 모두 포함)
- b. 비어 있지 않은 문자열, 비어 있지 않은 리스트, 비어 있지 않은 튜플, 비어 있지 않은 세트, 비어 있지 않은 딕셔너리
- c. True

2. False로 평가되는 주요 값

- a. False
- b. 0
- c. 빈 문자열
- d. 빈 리스트 [], 빈 튜플 (), 빈 세트 set(), 빈 딕셔너리 {}
- e. None

할당 연산자:

```
num = 10

num += 1    # num = num + 1
print(num)
num -= 1
print(num)
num *= 10
print(num)
num /= 10
print(num)
num %= 2
print(num)

11
10
100
10.0
0.0
```

- num = num + 1 이라고 생각하면 쉽다.

비트 연산자: 2진법 계산

비트 연산자는 정수를 이진 비트로 표현했을 때 사용하는 연산자들입니다. 각 연산자는 정수의 비트 단위로 동작합니다.

& : 비트 단위 AND

| : 비트 단위 OR

^ : 비트 단위 XOR

~ : 비트 단위 NOT

<< : 왼쪽 시프트

>> : 오른쪽 시프트

코딩을 시작하거나 시로 코드를 생성하세요.

- 60이 되기 위해서는 32, 16, 8, 4 가 필요하며 2, 1은 필요없다 (2^n)
- 따라서 111100이다.

비트 연산자는 정수를 이진 비트로 표현했을 때 사용하는 연산자들입니다. 각 연산자는 정수의 비트 단위로 동작합니다.

& : 비트 단위 AND

| : 비트 단위 OR

^ : 비트 단위 XOR

~ : 비트 단위 NOT

<< : 왼쪽 시프트

>> : 오른쪽 시프트

- 13이되기위해서는 8, 4, 1이 필요하다
- 1101

따라서 $(60 \& 13) = (8 + 4) = 12$ 이다.

8. 파이썬 연산자.ipynb

파일 수정 보기 실행 인터프리터 도구 도움말

변경어 + 코드 + 텍스트

4. 비트 연산자

비트 연산자는 정수를 이진 비트로 표현했을 때 사용하는 연산자들입니다. 각 연산자는 정수의 비트 단위로 동작합니다.

& : 비트 단위 AND

| : 비트 단위 OR

^ : 비트 단위 XOR

~ : 비트 단위 NOT

<< : 왼쪽 시프트

>> : 오른쪽 시프트

코딩을 시작하거나 시로 코드를 생성하세요.

AND

T T = T

F T = F

F F = F

60

2⁰

64 32 16 8 4 2 1

1 1 1 1 0 0 (60)

0 0 1 1 0 1 (13)

0 0 1 1 0 0 (12)

(60 | 13)

단위 AND

단위 OR

단위 XOR

단위 NOT

시프트

왼쪽 시프트

시작하거나 시로 코드를 생성하세요.

60

2°

64 32 16 8 4 2 1

1 1 1 1 0 0 (60)

0 0 1 1 0 1 (13)

0 0 1 1 0 0 (12)

1 1 1 1 0 1 (61)

(60 ^ 13)

4. 비트 연산자

F T = F

비트 연산자는 정수를 이진 비트로 표현했을 때 사용하는 연산자들입니다. 각 연산자는 정수의 비트 단위로 동작합니다.

& : 비트 단위 AND

| : 비트 단위 OR

^ : 비트 단위 XOR

~ : 비트 단위 NOT

<< : 왼쪽 시프트

>> : 오른쪽 시프트

코딩을 시작하거나 시로 코드를 생성하세요.

60

2°

64 32 16 8 4 2 1

1 1 1 1 0 0 (60)

0 0 1 1 0 1 (13)

0 0 1 1 0 0 (12)

1 1 0 0 0 1 (49)

```
a = 60 # 111100
b = 13 # 001101

print(a & b) # 12, 즉 1100
print(a | b) # 61, 즉 111101
print(a ^ b) # 49, 즉 110001
print(~a) # -61
print(a << 2) # 240, 즉 11110000
print(a >> 2) # 15, 즉 1111
```

12
61
49
-61
240
15

데이터의 크기를 나타내는 용량 단위

Bit: 가장 기본적인 데이터 단위로, 0 또는 1의 값을 가질 수 있습니다.

바이트(byte): 8개의 비트로 구성되며, 일반적으로 하나의 문자(알파벳, 숫자 등)를 저장할 수 있는 크기.

킬로바이트(KB): 1,024 바이트

메가바이트(MB): 1,024 KB (약 1,048,576 바이트)

기가바이트(GB): 1,024 MB (약 1,073,741,824 바이트)

테라바이트(TB): 1,024 GB (약 1,099,511,627,776 바이트)

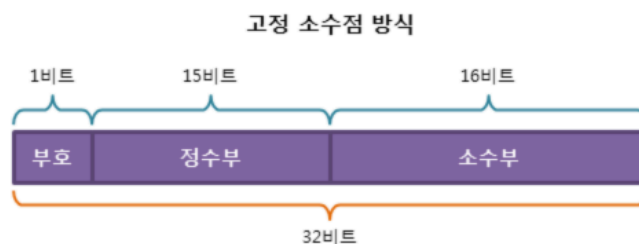
페타바이트(PB): 1,024 TB (약 1,125,899,906,842,624 바이트)

엑사바이트(EB): 1,024 PB (약 1,152,921,504,606,846,976 바이트)

제타바이트(ZB): 1,024 EB (약 1,180,591,620,717,411,303,424 바이트)

요타바이트(YB): 1,024 ZB (약 1,208,925,819,614,629,174,706,176 바이트)

1.1 + 0.1 == 1.2가 False로 나오는 이유?



부동소수점(Floating Point) 연산의 오차 때문입니다. 컴퓨터는 10진수를 2진수로 변환하여 계산하는데, 1.1과 0.1은 2진수로 정확히 표현되지 않아서 미세한 오차가 발생합니다. 1.1과 0.1 같은 10진수를 이진수로 변환할 때, 컴퓨터는 이를 무한 소수로 처리합니다. 이진수 표현에서는 특정 소수를 정확히 표현하지 못하고 근사값으로 저장하게 됩니다. 따라서 1.1 + 0.1의 결과는 1.2에 아주 가까운 값이지만, 정확히 1.2와는 다르기 때문에 False가 반환됩니다.

논리 연산자:

AND

```
# and
print(True and True)
print(False and True)
print(True and 3)      # 3
print(3 and 5)         # 5
print(0 and 5)         # 0
```

- True and 3 → 3: and 연산자는 모든 피연산자가 참일 때 마지막 참인 값을 반환합니다. True는 참이고, 3도 참(숫자 0이 아닌 값은 참으로 간주)이라서 3을 반환합니다.
- 3 and 5 → 5: 3과 5 모두 참이므로, 마지막 값인 5를 반환합니다.
- 0 and 5 → 0: 0은 거짓(0은 거짓으로 간주)이라서 and는 여기서 평가를 멈추고 0을 반환합니다. 5는 평가되지 않습니다.

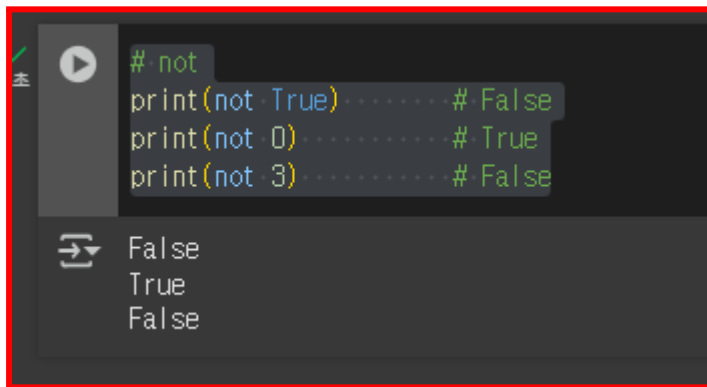
OR

```
# or
print(False or 3)      # 3
print(3 or 5)          # 3
print(0 or 5)          # 5
```

```
3
3
5
```

- False or 3 → 3: or 연산자는 첫 번째 참인 값을 반환합니다. False는 거짓이므로 다음 값인 3을 평가하고, 3은 참이므로 3을 반환합니다.
- 3 or 5 → 3: 3이 이미 참이므로 or는 여기서 평가를 멈추고 3을 반환합니다. 5는 평가되지 않습니다.
- 0 or 5 → 5: 0은 거짓이므로 다음 값인 5를 평가하고, 5는 참이므로 5를 반환합니다.

NOT

A screenshot of a Python REPL (Jupyter Notebook) showing the use of the 'not' operator. The code cell contains three lines: '# not', 'print(not True)# False', 'print(not 0)# True', and 'print(not 3)# False'. The output cell shows 'False', 'True', and 'False' on separate lines, corresponding to the three print statements.

```
# not
print(not True) .....# False
print(not 0) .....# True
print(not 3) .....# False
```

```
False
True
False
```

- not True → False: not은 부울 값을 반대로 만듭니다. 따라서 True의 반대는 False입니다.
- not 0 → True: 0은 거짓으로 간주되므로, not 0은 True를 반환합니다.
- not 3 → False: 3은 참으로 간주되므로, not 3은 False를 반환합니다.