

비트 연산자

- 값을 0과 1로 이뤄진 2진수, 즉 비트로 변환한 뒤에 비트 단위로 연산하는 연산자

연산자	사용 방법	의미
&	a & b	a 와 b 를 비트 AND 처리
	a b	a 와 b 를 비트 OR 처리
^	a ^ b	a 와 b 를 비트 XOR 처리
~	~a	a 를 비트 NOT 처리(보수 연산)
<<	a << 2	a 를 비트로 변환하여 왼쪽으로 2비트 이동
>>	a >> 2	a 를 비트로 변환하여 오른쪽으로 2비트 이동

비트연산자

X	Y	X & Y	X Y	X ^ Y
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

XOR 연산							
0 b	1	0	1	1	1	1	1
0 b	0	1	1	0	0	1	0

0 b	1	1	0	1	1	0	1

실습

```
a = 78 # 0b1001110
b = 85 # 0b1101000
print('{} : {}'.format(a, bin(a)))
print('{} : {}'.format(b, bin(b)))
result = a & b
print('a & b : {}'.format(result , bin(result)))
result = a | b
print('a | b : {}'.format(result , bin(result)))
result = a ^ b
print('a ^ b : {}'.format(result , bin(result)))
result = a << 1
print('a << 1 : {}'.format(result , bin(result)))
result = a >> 1
print('a >> 1 : {}'.format(result , bin(result)))
result = ~a
print('~a : {}'.format(result , bin(result)))
```

스퀀스 연산자

- 순서가 있는 데이터 구조인 스퀀스 객체(리스트, 튜플, range, 문자열 등)에서 사용되는 연산자

연산자	의미	사용 방법	결과
+	연결 연산자	'hello' + ' ' + 'python'	'hello python'
*	반복 연산자	'★' * 5	'★★★★★'

실습

```
course = 'programming'
subject = 'python'
name = 'Hong'

print('-' * 30)
print('Course : ' + course)
print('Subject : ' + subject)
print('Name : ' + name)
print('-' * 30)
```

멤버 연산자

- 어떤 값이 지정된 컬렉션(리스트, 문자열 등)에 속해 있는지 판단하는 연산자

연산자	의미	사용 방법	결과
in	포함되어 있으면 True	'python' in 'hello python'	True
		4 in [1, 2, 3, 4]	True
not in	포함되어 있지 않으면 True	'hi' not in 'hello android'	True
		5 not in [1, 2, 3, 4]	True

실습

```
comment = 'Hello Python!'

result = 'Python' in comment
print(f'결과 : {result}')

result = 'Java' not in comment
print(f'결과 : {result}')
```

연습 문제

- 네 자리 정수로 구성된 사원번호를 기준으로 근무 시간을 결정하고자 한다.
- 사원번호의 끝자리 숫자가 5 이상이면 ‘오전’ 아니면 ‘오후’를 출력하는
- 프로그램을 작성하세요.

연습 문제

- 한 변의 길이를 입력 받아 사각형을 그리는 프로그램을 작성하세요.
 - 한 변의 길이가 200 미만이면 빨간색 선의 사각형을 그린다.
 - 한 변의 길이가 200 이상이면 초록색 선의 사각형을 그린다.

연습 문제

- 2개의 정수를 입력 받아서 큰 값과 작은 값을 각각 출력하는 프로그램을 작성하세요.