

# **PYTHON\_PROGRAMMING**

## **연산자**



# 산술연산자

산술 연산자	사용 예	의 미
=	$a = b$	대입 연산자
+	$a + b$	더하기
-	$a - b$	빼기
*	$a * b$	곱하기
/	$a / b$	나누기
//	$a // b$	나누기(몫)
%	$a \% b$	나머지 값
**	$a ** b$	제곱

# 예제

```
num1 = 9; num2 = 2
```

```
print(num1 , " + " , num2 , " = " , num1 + num2)  
print(num1 , " - " , num2 , " = " , num1 - num2)  
print(num1 , " * " , num2 , " = " , num1 * num2)  
print(num1 , " / " , num2 , " = " , num1 / num2)  
print(num1 , " // " , num2 , " = " , num1 // num2)  
print(num1 , " % " , num2 , " = " , num1 % num2)  
print(num1 , " ** " , num2 , " = " , num1 ** num2)
```

# 관계 연산자

- ❖ 관계 연산자
- ❖ -. 왼쪽에 있는 값을 기준으로 두 값의 관계를 비교
- ❖ -. 조건을 만족하면 참(True), 조건을 만족하지 못하면 거짓(False)을 반환

관계 연산자	사용 예	의 미
<	$a < b$	a가 b 보다 작다
>	$a > b$	a가 b 보다 크다
<=	$a \leq b$	a가 b 보다 작거나 같다
>=	$a \geq b$	a가 b 보다 크거나 같다
==	$a == b$	a가 b 와 같다
!=	$a != b$	a가 b 와 같지 않다

# 예제

```
su1=3.1; su2=3
```

```
print("su1 >= su2 : ",(su1 >= su2))
```

```
print("su1 <= su2 : ",(su1 <= su2))
```

```
print("su1 == su2 : ",(su1 == su2))
```

```
print("su1 != su2 : ",(su1 != su2))
```

# 대입 연산자

대입 연산자	사용 예	의 미
<code>+=</code>	<code>a += b</code>	<code>a = a + b</code>
<code>-=</code>	<code>a -= b</code>	<code>a = a - b</code>
<code>*=</code>	<code>a *= b</code>	<code>a = a * b</code>
<code>/=</code>	<code>a /= b</code>	<code>a = a / b</code>
<code>//=</code>	<code>a //= b</code>	<code>a = a // b</code>
<code>%=</code>	<code>a %= b</code>	<code>a = a % b</code>
<code>**=</code>	<code>a **= b</code>	<code>a = a ** b</code>

# 예제

```
su1 = su2 = 5
```

```
su1+=1
```

```
print("su1 + 1 = ",su1)
```

```
su1-=1
```

```
print("su1 - 1 = ",su1)
```

```
su1*=su2
```

```
print("su1 * su2 = ",su1)
```

```
su1//=su2
```

```
print("su1 // su2 = ",su1)
```

```
su1%=su2
```

```
print("su1 % su2 = ",su1)
```

# 예제

```
su1 = 5
```

```
su2 = 3
```

```
su1**=su2
```

```
su1-=2
```

```
print("su1 / 4 = ",su1 / 4)
```

```
print("su1 // 4 = ",su1 // 4)
```

```
print("su1 % 4 = ",su1 % 4)
```



# 논리 연산자

## ❖ 참과 거짓을 판별하는 연산

논리 연산자	사용 예	의 미
and	$(a > b) \text{ and } (a < c)$	a가 b보다 크고 a가 c보다 작으면 참
or	$(a > b) \text{ or } (a < c)$	a가 b보다 크거나 a가 c보다 작으면 참
not	$\text{not}(a == b)$	a가 b보다 크면 거짓

A	B	A or B	A and B
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

# 예제

```
print(0 or 0," : ",False or False)
```

```
print(1 or 0," : ",True or False)
```

```
print(0 or 1," : ",False or True)
```

```
print(1 or 1," : ",True or True)
```

```
print("not : ",not(0 or 0)," : ",not(False or False))
```

```
print("not : ",not(1 or 1)," : ",not(True or True))
```

# 예제

```
print(0 and 0," : ",False and False)
```

```
print(1 and 0," : ",True and False)
```

```
print(0 and 1," : ",False and True)
```

```
print(1 and 1," : ",True and True)
```

```
print("not : ",not 0," : ",not False)
```

```
print("not : ",not 1," : ",not True)
```

# 비트 연산자

- ❖ 비트 연산자
- ❖ -. 2진수로 변환하여 비트 단위의 연산을 수행하는 연산자

논리 연산자	사용 예	의 미
	$a \mid b$	a와 b를 bit로 변환하여 OR 연산
&	$a \& b$	a와 b를 bit로 변환하여 AND 연산
^	$a \wedge b$	a와 b를 bit로 변환하여 XOR 연산
~	$\sim a$	a를 bit로 변환하여 NOT 연산
>>	$a \gg 2$	a를 bit로 변환하여 오른쪽으로 Shift
<<	$a \ll 2$	a를 bit로 변환하여 왼쪽으로 Shift

# 예제

```
num1 = 3
```

```
num2 = 5
```

```
result = num1 | num2
```

```
print(result)
```

# 예제

```
num1 = 3
```

```
num2 = 5
```

```
result = num1 & num2
```

```
print(result)
```

# 예제

```
num1 = 3
```

```
num2 = 5
```

```
result = num1 ^ num2
```

```
print(result)
```

# 연산자 우선순위

우선순위	연산자	설명
1	() [] {}	괄호, 리스트, 딕셔너리, 세트 등
2	**	지수
3	+ - ~	단항 연산자
4	* / % //	산술 연산자
5	+ -	산술 연산자
6	<< >>	비트 시프트 연산자
7	&	비트 논리곱
8	^	비트 배타적 논리합
9		비트 논리합
10	< > >= <=	관계 연산자
11	== !=	동등 연산자
12	= %= /= //= -= += *= **=	대입 연산자
13,14,15	not, and, or	논리 연산자
16	if ~else	비교식