



컴퓨팅사고와 코딩기초

Chapter 8. 연산자

목차

- 연산자
- 산술연산자
- 할당연산자
- 비교연산자
- 논리연산자

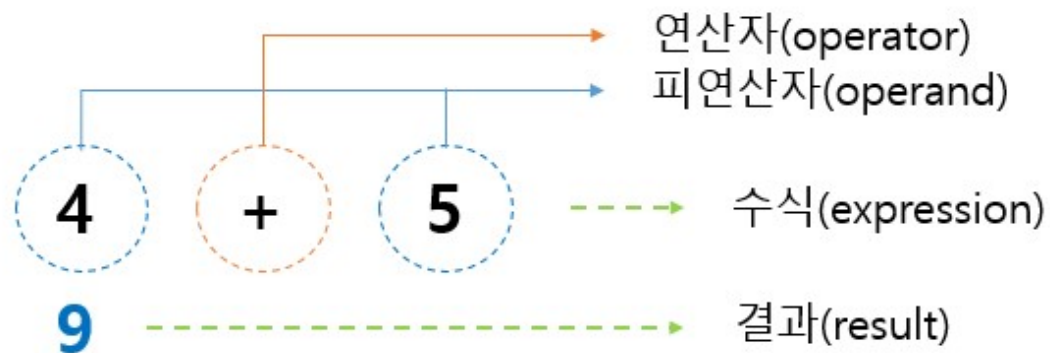


연산자

- 수식과 연산자

- 파이썬 IDLE를 실행하여 파이썬 셸 창에서 정수 계산식 입력
- 파이썬 셸 창에 입력된 수식은 연산자와 피연산자로 구성
- 연산에 의한 결과(result) 바로 출력

```
>>> 4 + 5
9
>>> 4 - 5
-1
>>> 4 * 5
20
>>> 4 / 5
0.8
```



연산자

- 연산자 종류
 - 산술 연산자(Arithmetic Operators)
 - 할당 연산자(Assignment Operators)
 - 비교 연산자(Comparison Operators)
 - 논리 연산자(Logical Operators)
 - 항등 연산자(Identity Operators)
 - 멤버 연산자(Membership Operators)
 - 비트 연산자(Bitwise Operators) 등
-

연산자

- 연산자 종류

산술 연산자	할당 연산자	비교 연산자	논리 연산자
+	=	==	and
-	+=	!=	or
*	-=	<>	not
/	*=	>	
**	/=	<	
//	%=	>=	
%	**=	<=	
	//=		

- 멤버 연산자로 in, not in 등
- 항등 연산자(Identity Operators)로 is, not is 등

산술 연산자

- 정수의 연산

사칙연산	연산자	수식	결과	결과 자료형
덧셈	+	>>>8 + 5	13	정수(integer)
뺄셈	-	>>>8 - 5	3	정수(integer)
곱셈	*	>>>8 * 5	40	정수(integer)
나눗셈	/	>>>8 / 5	1.6	실수(float)

산술 연산자

- 정수의 연산

- 나누기와 관련된 산술연산에서 피연산자가 모두 정수의 경우 몫을 구하는 연산의 결과는 정수
 - 나머지를 구하는 연산(%)에서도 결과는 정수
 - 정수 0이 피젯수로 사용되면 항상 결과 0, 결과 type은 실수가 되므로 0.0으로 출력
 - 반대로 0이 제수로 사용되면 모든 수를 0으로 나눌 수 없으므로 에러가 발생
-

산술 연산자

- 실수의 연산
 - 피연산자가 실수인 경우 결과도 모두 실수(float)로 출력
 - 실수를 부동소수점(float) 방식으로 일부 계산에서 오차값 출력
 - 실수 연산에서 오차값의 문제를 해결하는 방법으로 decimal 내장 모듈을 사용할 수 있음

산술 연산자

- 문자열의 연산

- 문자열을 피연산자로 하여 산술 연산을 할 수 있음
- 사용할 수 있는 산술 연산자는 '더하기(+)'와 '곱하기(*)'
- 문자열 더하기(+) 연산
 - 피연산자는 모두 '문자열'만 가능하고 숫자는 사용할 수 없음
- 문자열 곱하기(*) 연산
 - 연산의 피연산자는 '문자열'과 '정수' 사용
 - 문자열에 정수를 곱하면 문자열이 곱하는 정수만큼 반복되어 출력

할당 연산자

- 연산식에서 필요에 따라 값을 바꿀 수 있는 '변수'를 피연산자로 사용할 수도 있음
 - 변수에 특정한 값이 할당(대입)되기 위해 사용하는 연산자 '='을 할당 연산자(assignment operator)라 함
 - 할당문(대입문)의 기본 형식은 할당 연산자를 가운데 두고, 왼쪽에는 변수가 오고 오른쪽에는 상수, 변수, 수식
-

할당 연산자

- 다중 할당문

- 하나의 할당문을 작성하여 여러 개의 변수에 값을 할당하는 것 가능
 - 여러 개의 변수에 다른 값을 할당하는 다중 할당문
 - 두 수의 교환을 복잡한 알고리즘을 사용하지 않고 다중 할당문 하나로 즉시 수행할 수도 있음
-

할당 연산자

- 복합 할당문

할당연산자	기능	의미
<code>+=</code>	왼쪽 변수에 오른쪽 값을 더하고 그 결과를 왼쪽 변수에 할당한다.	$a += b$ 는 $a = a + b$ 를 의미함
<code>-=</code>	왼쪽 변수에 오른쪽 값을 빼고 그 결과를 왼쪽 변수에 할당한다.	$a -= b$ 는 $a = a - b$ 를 의미함
<code>*=</code>	왼쪽 변수에 오른쪽 값을 곱하고 그 결과를 왼쪽 변수에 할당한다.	$a *= b$ 는 $a = a * b$ 를 의미함
<code>/=</code>	왼쪽 변수에 오른쪽 값을 나누고 그 결과를 왼쪽 변수에 할당한다.	$a /= b$ 는 $a = a / b$ 를 의미함
<code>%=</code>	왼쪽 변수에 오른쪽 값을 나눈 후 그 나머지를 왼쪽 변수에 할당한다.	$a \% = b$ 는 $a = a \% b$ 를 의미함
<code>//=</code>	왼쪽 변수에 오른쪽 값을 나눈 후 그 몫을 왼쪽 변수에 할당한다.	$a //= b$ 는 $a = a // b$ 를 의미함
<code>**=</code>	왼쪽 변수에 오른쪽 값을 제곱하고 그 결과를 왼쪽 변수에 할당한다.	$a ** = b$ 는 $a = a ** b$ 를 의미함

비교 연산자

- 두 개의 피연산자의 값이 서로 같은지, 다른지, 큰지, 작은지 등 판단, 결과를 참(True) 또는 거짓(False)으로 산출해 주는 연산자
 - 비교 연산자는 뒤에서 배우는 논리 연산자와 함께 파이썬의 조건문(if), 반복문(while) 구문에 자주 사용되므로 잘 활용할 수 있어야 함
-

비교 연산자

- 비교 연산자

종류	의미	결과 예시 (a:7, b:3의 경우)
==	양쪽 값이 같다	a == b 는
		False
!=	양쪽 값이 같지 않다	a != b 는
		True
>	왼쪽 값이 크다	a > b 는
		True
<	왼쪽 값이 작다	a < b 는
		False
>=	왼쪽 값이 크거나 같다	a >= b 는
		True
<=	왼쪽 값이 작거나 같다	a <= b 는
		False

비교 연산자

- 항등 연산자와 차이
 - 비교 연산자 중 '=='와 '!='와 유사한 연산자로 항등 연산자(Identity Operators)인 is, not is 등이 있음
 - '양쪽 값이 같다(is)'와 '양쪽의 값이 같지 않다(not is)' 의미의 연산자이나 중요한 다른 차이가 있음
 - 비교 연산자(==, !=)은 **값의 크기** 비교
 - 항등 연산자(is, is not)는 **메모리에 저장된 객체(object)** 비교
-

논리 연산자

- 논리 연산자는 여러 개의 비교 연산식 또는 비교 연산의 결과(True, False)를 연결하여 전체적으로 참(True) 또는 거짓(False)을 판단해 그 결과를 산출해 주는 연산자

논리 연산자	문법	설명
and	a and b	AND(논리곱), 양쪽 모두 참일 때 참
or	a or b	OR(논리합), 양쪽 중 한쪽만 참이라도 참
not	not x	NOT(논리부정), 참과 거짓을 뒤집음

논리 연산자

- 논리 연산자의 단락 평가

False and True









→ 앞 부분이 False and 이므로 거짓(False)이 확정된 상태이기 때문에 뒤에 있는 값은 확인하지 않고 False 반환

True or False

→ 앞 부분이 True or 이므로 참(True)이 확정된 상태이기 때문에 뒤에 있는 값은 확인하지 않고 True 반환

논리 연산자

- 논리 연산과 논리 게이트
 - 불 연산 논리 게이트

게이트	기호	의미	진리표	논리식															
AND		입력신호가 모두 1일 때 1출력	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1	$Y = A \cdot B$ $Y = AB$
A	B	Y																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OR		입력신호 중 1개만 1이어도 1출력	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1	$Y = A + B$
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
NOT		입력된 정보를 반대로 변환하여 출력	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0	$Y = A'$ $Y = \overline{A}$									
A	Y																		
0	1																		
1	0																		
BUFFER		입력된 정보를 그대로 출력	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	Y	0	0	1	1	$Y = A$									
A	Y																		
0	0																		
1	1																		
NAND		NOT + AND, 즉 AND의 부정	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0	$Y = \overline{A \cdot B}$ $Y = \overline{AB}$
A	B	Y																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
NOR		NOT + OR, 즉 OR의 부정	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	0	$Y = \overline{A + B}$
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
XOR		입력신호가 모두 같으면 0, 한 개라도 틀리면 1출력	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0	$Y = A \oplus B$ $Y = \overline{AB} + A\overline{B}$
A	B	Y																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
XNOR		NOT + XOR, 즉 XOR의 부정	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1	$Y = A \odot B$ $Y = A \oplus \overline{B}$ $Y = AB + \overline{A}\overline{B}$
A	B	Y																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	1																	