

[기본 연산자]

■ 항과 연산자

- 단항 연산자: 증가 감소 연산자
- 이항 연산자: 사칙 연산자
- 삼항 연산자: 조건 연산자
- 연산자 우선순위: 단항 > 이항 > 삼항

■ 대입 연산자

- 왼쪽 변수 lValue = 오른쪽 변수 rValue
- 변수에 값을 대입하는 연산자
- 이항 연산자 중 우선순위가 가장 낮음

■ 부호 연산자

- +, - 두가지
- 이항 연산자: 더하기, 빼기
- 단항 연산자: 양수, 음수

■ 산술 연산자

- 덧셈, 뺄셈, 곱셈, 나눗셈, 나머지
- 곱셈: x 가 아닌 *
- 나눗셈: ÷ 가 아닌 /
- 우선순위: 일반 산술 연산과 같으며 %는 *, / 과 같음

■ 증가 감소 연산자

- 연산자 앞뒤에 사용하여 1 만큼 증가 감소
- ++/--가 변수 앞에: 변수값이 증감된 후 대입
-> 문장 끝나기 (;) 전에 변수값 증감
- ++/--가 변수 뒤에: 변수값이 대입된 후 증감
-> 문장 끝난 (;) 후에 변수값 증감

■ 관계 연산자

- 부등호(비교 연산자), 이항 연산자
- 결과값은 참(true) 또는 거짓(false)으로 반환
- >, <, >=, <=, ==, !=
- 우선순위: 관계 연산자 > 대입 연산자
- 조건식이나 반복문을 제어하는데 주로 사용

■ 논리 연산자

- 논리 곱 &&: 모두 참인 경우 만 참
- 논리 합 ||: 모두 거짓인 경우 만 거짓
- 부정 |: 참 <-> 거짓
- 우선순위: 관계 연산자 > 논리 연산자
- 단락 회로 평가: 논리곱 논리합 연산을 할 때 두 항을 모두 실행 않더라도 결과값을 알 수 있는 경우 나머지 항은 실행되지 않는 것
-> 예상한 결과와 다를 수 있으므로 주의

■ 복합 대입 연산자

- 산술 연산자, 비트 연산자를 함께 사용하여 코드를 간결하게 표현
- +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, |=, ^=
- >>=: 왼쪽에 채워지는 비트값은 부호비트와 동일

- >>>=: 왼쪽에 채워지는 비트값은 "0"

■ 조건 연산자

- 조건식? 결과 1 : 결과 2
- 조건식이 참이면 결과 1, 거짓이면 결과 2 가 선택

[비트 연산자]

■ 비트 논리 연산자

- & (and) 연산자: 2 개의 비트값이 모두 1 인 경우 결과가 1
- | (or) 연산자: 비트값이 1 개라도 1 인 경우 결과가 1
- ^ (xor) 연산자: 2 개의 비트값이 같은 값이면 0, 다른 값이면 1
- ~ (반전) 연산자: 비트값을 0 은 1 로, 1 은 0 으로 바꿈

■ 비트 이동 연산자 (시프트 연산자)

- << n 연산자: 왼쪽으로 n 비트 이동
- >> n 연산자: 오른쪽으로 n 비트 이동
- >>> n 연산자: 오른쪽으로 n 비트 이동하되, 왼쪽에 채워지는 비트값은 0

■ 연산자 우선순위

- 단항 > 이항 > 삼항
- 대입 연산자가 가장 낮다
- 산술 > 관계 > 논리 > 대입
- ()의 우선순위가 가장 높다