

1. 데이터 타입

정수 타입	byte, short, int, long
문자 타입	char
실수 타입	float, double
논리 타입	boolean

2. 데이터 크기 : 8비트에서 64비트로 갈수록 커짐 (8 < 16 < 32 < 64)

8비트	byte, boolean
16비트	short, char
32비트	int, float
64비트	long, double

3. 데이터 타입 변환

데이터 타입을 바꾸는 연산

- 자동 타입 변환 : 큰 데이터 타입으로 작은 데이터 타입을 통일한다.

```
double d1 = 5 * 3.14;  정수 5를 실수 5.0 으로 자동 타입 변환  
double d2 = 1;        정수 1을 실수 1.0 으로 자동 타입 변환
```

- 강제 타입 변환 : 데이터 손실이 발생할 수 있다.

```
float f = ( float ) 3.14;  double 의 3.14 를 float 로 형 변환해 f 에 3.14F 저장  
byte b = ( byte ) 300;    int 의 300을 byte로 형 변환하면 데이터 손실 발생  
byte x = ( byte ) 3.14;   double의 3.14를 byte 로 형 변환하면 데이터가 손실되고 3만 저장  
double d = ( double ) 3.14F; float 의 3.14를 double 로 형 변환하면 데이터 손실 없이 저장
```

4. 데이터 입력 받기

```
package java;

import java.util.Scanner;  // 입력을 받기 위해선 필수로 입력

public class Input {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);  // 입력
        Int num = sc.nextInt();  // 입력받은 sc 를 num 에 저장
        System.out.println(num);  // 저장된 num 을 출력
    }
}
```

5. 연산자 : 어떠한 기능 또는 어떤 대상체에 계산과 같은 처리를 수행하는 문자 또는 기호

비교 논리 연산자	>, <, >=, <=, ==, !=
대입 연산자	=, +=, -=, *=, /=, &=, !=
부호, 증감 연산자	++, --, +, -
조건 연산자	(expr) ? x : y

- 연산자(operator) : 어떠한 기능을 수행하는 기호 (ex +, -, *, / 등)
연산자는 크게 단항, 이항, 삼항, 대입 연산자
- 피연산자(operand) : 연산자의 작업 대상 (ex 변수, 상수, 수식 등)
이항 연산자는 산술, 비교, 논리 연산자

6. 연산자 우선순위

우선순위	연산자	피연산자	연산 방향
0	() 괄호 속 연산자	다양	-
1	증감 (++, --), 부호 (+, -), 비트 (~), 논리 (!)	단항	←
2	산술 (*, /%)	이항	→
3	산술 (+, -)	이항	→
4	쉬프트 (>>, <<, >>>)	이항	→
5	비교 (<, >, <=, >=, instanceof)	이항	→
6	비교 (==, !=)	이항	→
7	논리 &	이항(단항)	→
8	논리 ^	이항(단항)	→
9	논리	이항(단항)	→
10	논리 &&	이항	→
11	논리	이항	→
12	조건 (?:)	삼항	→
	대입		
13	(=, +=, -=, *=, /=, %=, &=, ^=, =, <<=, >>=, >>>=)	이항	←

- 연산자의 연산 진행방향은 왼쪽에서 오른쪽으로 수행
- 단항 연산자와 대입 연산자의 경우에는 오른쪽에서 왼쪽으로 수행
- 괄호의 우선순위가 제일 높고, 산술 > 비교 > 논리 > 대입의 순서
- 단항 > 이항 > 삼항의 순서