

1bit-> 0, 1

1byte=8bit -> 256 $-2^7 \sim +2^7 - 1$ -> $-128 \sim +127$

4byte=32bit

00000000

11111111

00000000 00000000 00000000 00000000

11111111 11111111 11111111 11111111

**** 컴퓨터 0, 1

사람 소스(자바.java) 컴파일 컴퓨터 0 1

10진수 2진수

10 1010(2)

byte b=10; // 1byte

00001010

int i=10; // 4byte

00000000 00000000 00000000 00001010

-> byte

00001010

b=(byte)i; // 큰자료형에서 작은 자료형으로 강제형변환
// 큰자료형의 데이터가 손실이 발생

i=b; // 작은자료형에서 큰자료형 형변환은 자동으로 변환
// 데이터 손실이 발생 하지 않는다.

00001010 -> 00000000 00000000 00000000 00001010

*** 기본 정수형 타입 -> int, 기본 실수형 타입 -> double

int i=10;

long l=10+i; // int+int-> int

byte b2=10;

int i2=10;

```
b2=b2+i2; // byte+int-> int+int-> int
byte=int;// 에러
```

***형변환의 규칙

1. 같은 자료형의 연산은 같은 형으로 형변환(기본)
2. 작은 자료형에서 큰자료형 변환시 자동형변환

```
int i=10;
long l=10+i;// int+int-> int
```

3. 작은자료형형과 큰자료형과의 연산 시 큰자료형으로 자동형변환
- ```
byte b2=10;
int i2=10;
```

```
b2=b2+i2; // byte+int-> int+int-> int
byte=int;// 에러
```

4. int형 아래 자료형으로 연산자 int형으로 자동 형변환  
-> short, byte

```
byte b1=10;
byte b2=20;
byte b3=b1+b2;// byte+byte-> int+int-> int (error)
byte b3=(byte)b1+b2;
byte b3=(byte)b1+(byte)b2;
```

```
byte b3=(byte)(b1+b2); // 강제 형변환 (byte)()
```

-----

#### \*\*\*형변환(타입변환)

>한 타입의 값을 다른 타입의 값으로 변환

##### 자동 타입 변환

- >컴파일러에 의해 원래의 타입보다 큰 타입으로 자동 변환
- >치환문(=)이나 수식 내에서 타입이 일치하지 않을때

##### 강제 타입 변환

- >개발자가 의도적 타입 변환
- >()안에 개발자가 명시적으로 타입 변환 지정
- >자료(데이터)의 손실이 발생

-----

```
byte b=127;// -2^7~+2^7-1 (-128~+127)
```

```
System.out.println(b);
```

```
// -128~+127
```

```
//-> 최대값위의 범위로 벗어나는(커지는) 경우->오버플로워
```

```
//-> 최소값아래의 범위로 벗어나는(작아지는) 경우-> 언더플로워
```

```
b=(byte)(b+1);// 127+1 -> 127보다 1크면 최소값(-128)로 변환
```

```
System.out.println(b);
```

```
b=(byte)129;// 127+1+1 -> 127보다 2크면 -127로 변환
```

```
System.out.println(b);
```

```
b=(byte)-128;
```

```
System.out.println(b);
```

```
b=(byte)-129;// -128-1 -> -128보다 1작으면 최댓값(+127) 변환
```

```
System.out.println(b);
```

```
b=(byte)-130;// -128-1-1 ->-128보다 2작으면 +126 변환
```

```
System.out.println(b);
```