

# 프로그래밍 언어 응용

chapter02

## 변수와 자료형

제공된 자료는 훈련생의 수업을 돕기 위한 것으로, 타인과 공유하시면 안됩니다.

# Contents

part.1

컴퓨터에서 데이터를 어떻게 표현할까?

part.2

변수

part.3

자료형

part.4

상수와 리터럴

part.5

형변환

# 컴퓨터에서 데이터를 어떻게 표현할까?

## 비트와 바이트

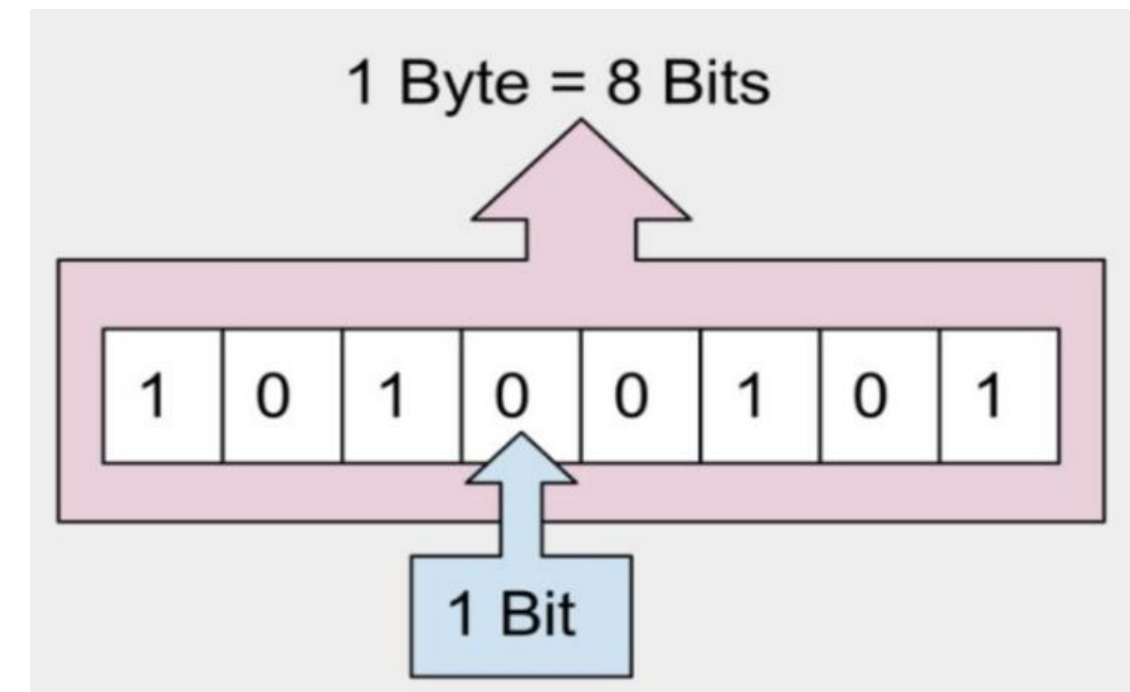
### 이진법과 비트(BIT)

- 컴퓨터는 이진 숫자 체계를 사용하여 숫자 0과 1로 값을 나타낸다.
- 비트는 데이터를 표현하는 가장 작은 단위이다.

### 바이트 (BYTE)

- 8개의 비트로 구성된다.
- 바이트는 비트의 묶음이다.

$$\begin{matrix} 2^2 & 2^1 & 2^0 \\ 1 & 1 & 1 \end{matrix}$$



# 컴퓨터에서 데이터를 어떻게 표현할까?

## 2진수와 10진수

### 10진수

- 일상생활에서 사용하는 숫자 체계이다.

### 2진수

- 컴퓨터 내부에서 사용하는 숫자 체계이다.

### 비트로 표현할 수 있는 수의 범위

자리수	값의 범위
1개	0,1 (0~1)
2개	00,01,10,11 (0~3)
3개	000,001,010,011,100,101,110,111 (0~7)

10진수	2진수
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
⋮	⋮

자동차의 정보를 프로그램으로 작성할 때는, 자동차의 다양한 특성을 변수로 표현할 수 있다.



자동차



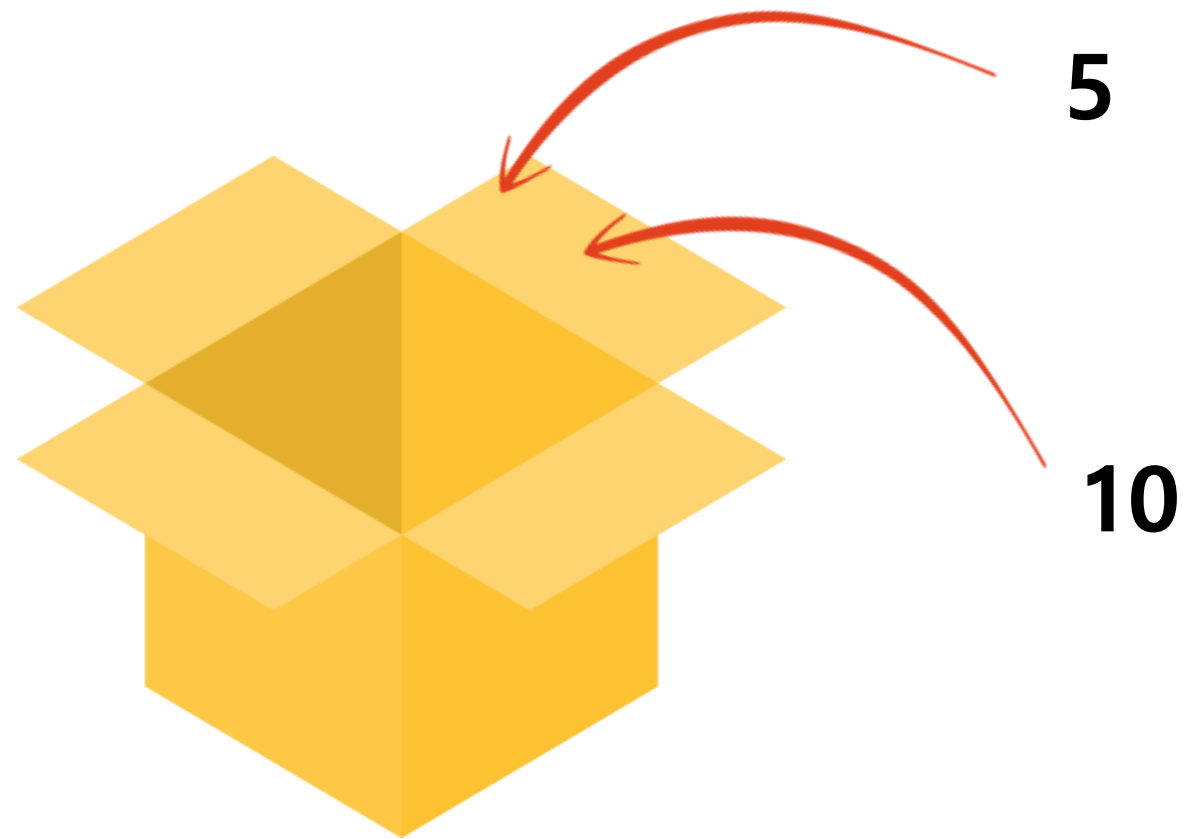
```
public class Car {  
    String model;  
    String brand;  
    int price;  
    String color;  
}
```

자동차 관리 프로그램

이해

변수

값을 저장하는 공간이며, 값은 변경될 수 있다



### 변수 선언하기

- 변수를 선언할 때는 먼저 자료형을 지정하고, 변수의 이름을 작성한다.

### 변수의 이름 짓기

- 변수의 이름은 용도에 맞게 작성한다.
- 변수의 이름은 소문자로 시작한다.
- 변수의 이름을 여러 단어로 조합할 경우에는 중간에 첫 글자를 대문자로 한다.

int a;


자료형

변수이름

### 변수에 값 대입하기

- 대입연산자를 쓰고 저장하고 싶은 값을 작성한다.
- 오른쪽 값이 왼쪽 변수에 저장된다.

a = 4;





이해

자료형

변수에 저장할 수 있는 값의 종류와 값의 범위

10 20 30



a b c



안녕하세요



기초자료형

- 자바에서 기본적으로 제공하는 타입이다.
- 기초자료형은 변수가 실제 값을 저장한다.
- 기초자료형은 고정된 크기를 가진다.

기초변수

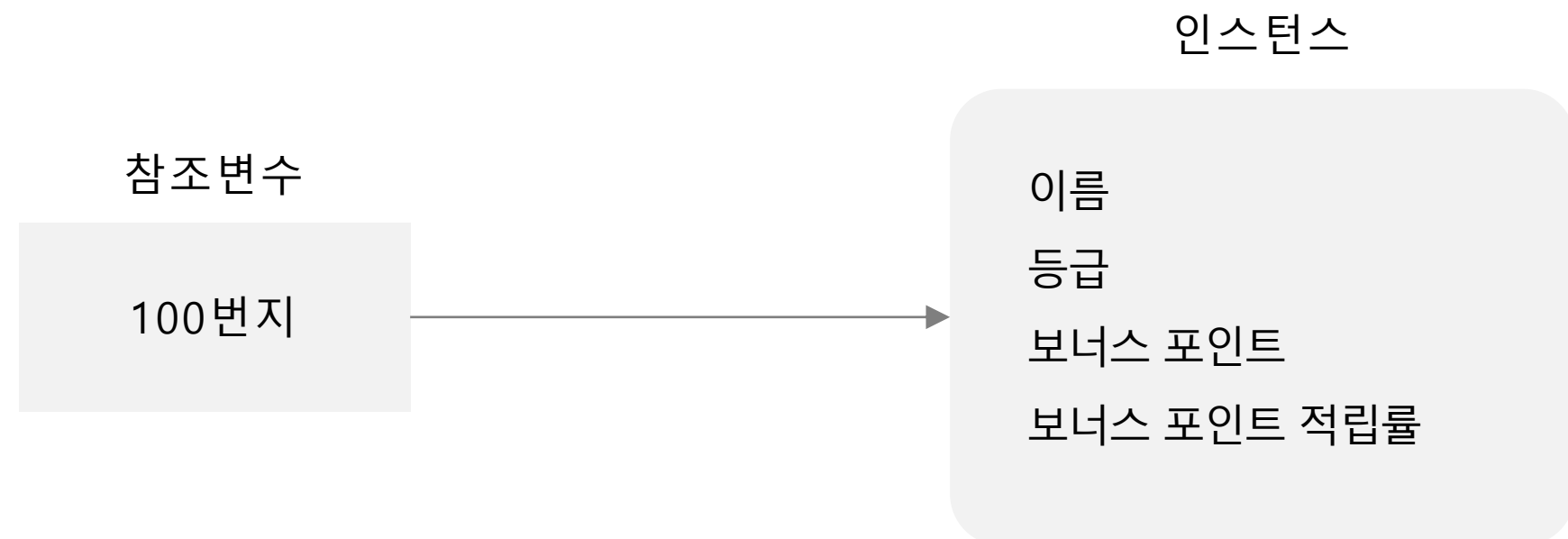
100

자료형의 종류

종류	자료형	크기	예시
정수	byte, short, int, long	1, 2, 4, 8	10, 20
실수	float, double	4, 8	23.4f, 34.567
문자	char	2	'A', 'a'
논리	boolean	1	true(참), false(거짓)

**참조자료형**

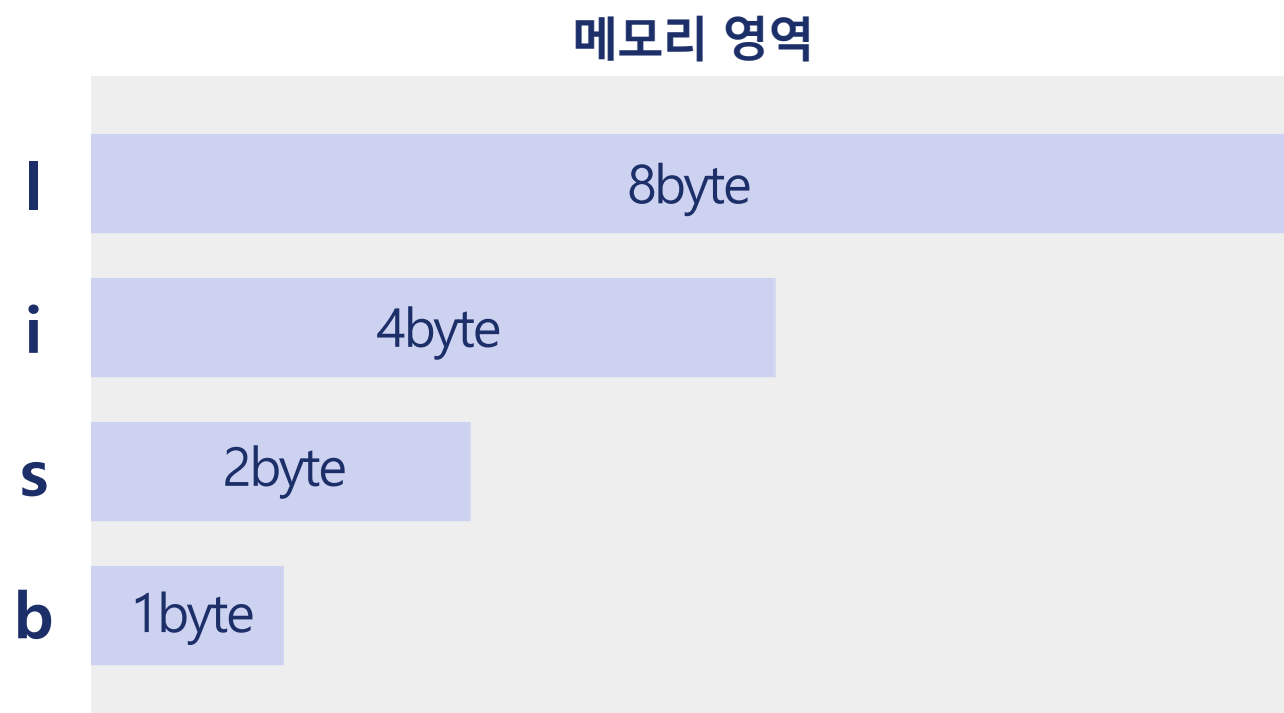
- 변수가 값 대신에 메모리 위치를 참조한다.
- 복잡한 데이터 구조를 다룰 때 사용한다.



### 정수 자료형의 특징

- 정수자료형은 양수, 음수, 0을 나타낼 수 있다.
- 정수자료형에는 바이트 크기별로 byte, short, int, long형 4종류가 있다.
- 정수 자료형 중 가장 많이 사용하는 것은 int형이다.

```
int i = 10;
```



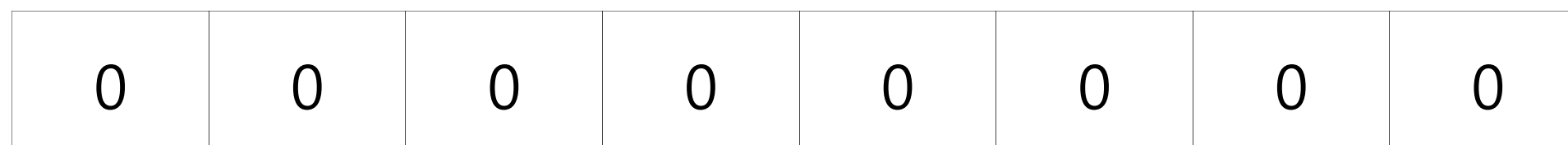
Q. 왜 숫자를 표현하는 방법이 4가지나 될까?

Q. 만약 1~100 범위의 숫자를 1000개 저장해야 한다면, 어떤 자료형을 사용해야 할까?

### 정수 자료형의 범위

- 정수 자료형의 첫 번째 자리는 부호비트이다.
- 0이면 양수, 1이면 음수를 의미한다.
- 예를 들어 byte형 변수는 부호비트를 제외한 나머지 7비트로 -128~127까지 표현할 수 있다.
- 만약, 범위에 벗어나는 값을 대입하면 오류가 발생한다.

```
byte b = 200;
```



부호비트

실제로 값을 표현하는 것은 7비트

# 자료형

## 문자 자료형

---

### 문자형

- 문자자료형은 문자 하나를 나타낼 수 있다.
- ' ' 작은 따옴표로 값을 감싸주어야 한다.

### 문자열형

- String은 문자열을 저장 할 수 있는 참조자료형이다.
- " " 큰 따옴표로 값을 감싸주어야 한다.

```
char ch = 'a';
```

```
String str = "안녕하세요";
```

## 인코딩이란?

- 컴퓨터는 문자를 이해할 수 없기 때문에 문자도 숫자로 변환해서 저장해야 한다.
- 인코딩은 숫자와 문자를 맵핑한 코드체계이고, 문자를 숫자코드로 변환하는 것을 문자인코딩이라고 한다.
- 예를 들어 사용자가 문자 A를 입력하면 문자에 해당하는 65라는 코드값이 저장된다.

Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
64	40	100	&#64;	@	96	60	140	&#96;	`
65	41	101	&#65;	A	97	61	141	&#97;	a
66	42	102	&#66;	B	98	62	142	&#98;	b
67	43	103	&#67;	C	99	63	143	&#99;	c
68	44	104	&#68;	D	100	64	144	&#100;	d
69	45	105	&#69;	E	101	65	145	&#101;	e
70	46	106	&#70;	F	102	66	146	&#102;	f
71	47	107	&#71;	G	103	67	147	&#103;	g
72	48	110	&#72;	H	104	68	150	&#104;	h
73	49	111	&#73;	I	105	69	151	&#105;	i
74	4A	112	&#74;	J	106	6A	152	&#106;	j
75	4B	113	&#75;	K	107	6B	153	&#107;	k
76	4C	114	&#76;	L	108	6C	154	&#108;	l
77	4D	115	&#77;	M	109	6D	155	&#109;	m

아스키코드를 보면 "7비트체계"로 128가지의  
문자를 표현할 수 있다

'A' → 65

영문 -> 아스키코드

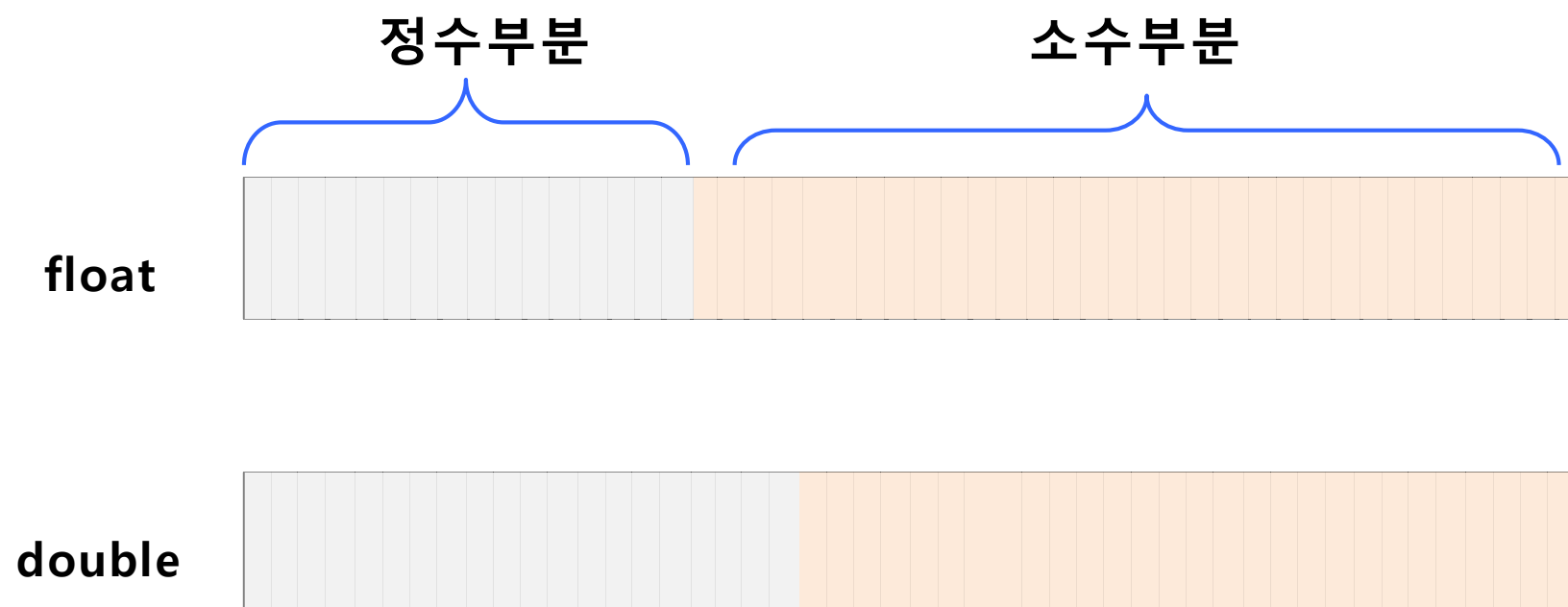
한글, 한자 ... -> UTF-8, UTF-16, EUC-KR ...

한글이나 한자를 표현하기 위해  
다양한 코드체계가 만들어졌다

### 실수 자료형

- 실수 자료형은 정수부분과 소수부분을 따로 표현한다.
- 예를 들어 3.14를 표현한다면 3이라는 정수부분과 .14라는 소수부분을 따로 표현한다.
- 실수 자료형에는 float형과 double형이 있다.

`float f = 3.14f;`



double형이 범위가 더 크기 때문에  
소수부분을 더 정밀하게 표현할 수 있다



### 논리형

- 논리자료형은 참, 거짓 값을 나타낼 때 사용한다.
- true는 참, false는 거짓을 의미한다.
- 결혼을 했는지? 값이 존재하는지? 와 같은 여부를 나타낼 때 주로 사용한다.

### 네이밍 규칙

- 변수의 이름은 일반적으로 is~로 시작한다.

```
boolean isMarried = false;
```

### 상수

- 상수는 프로그램에서 변하지 않는 수를 만들 때 사용한다.
- 상수는 한번 선언하면 변경할 수 없다.
- 예를 들어 원주율이나 1년은 12개월이라는 값 등이 있다.

### 리터럴

- 리터럴은 프로그램에서 사용하는 모든숫자, 문자, 논리값을 말한다.

```
final double PI = 3.14;
```

상수 키워드

```
int i = 10;
```

```
char ch = 'a';
```

```
boolean b = true;
```

# 형 변환

## 형 변환이란?

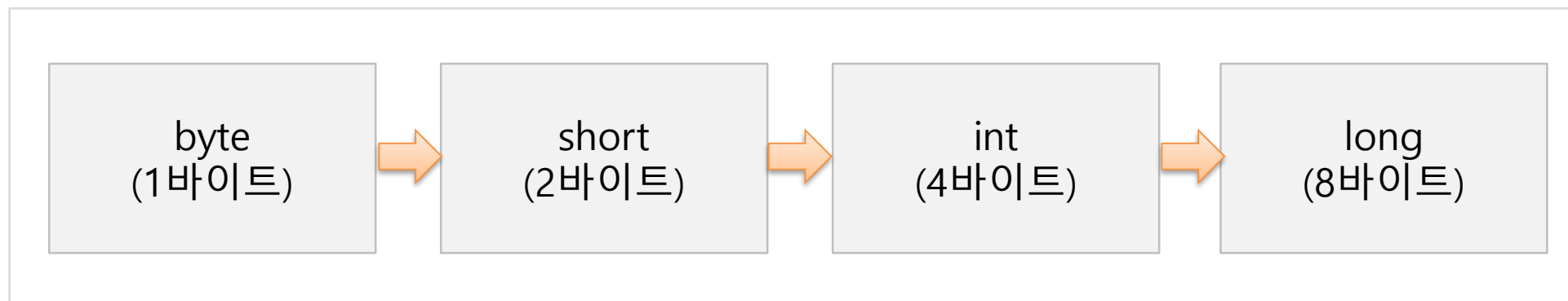
### 형 변환이란?

- 형 변환은 자료형을 다른 자료형으로 변환하는 것을 말한다.

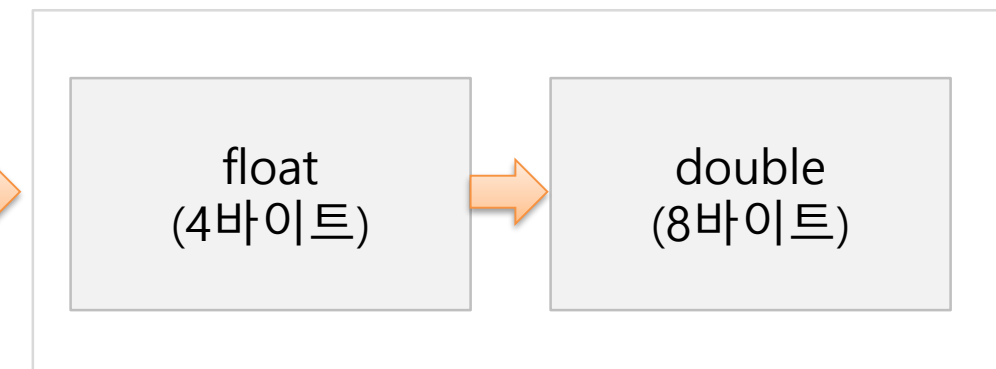
### 형 변환의 특징

- 형 변환은 크게 자동형 변환과 강제형 변환으로 구분한다.
- 바이트크기가 더크거나, 더정밀한 자료형으로 바꿀 때는 형 변환이 자동으로 이루어진다.
- 반대로 형 변환을 하려면 강제로 변환해야 하고, 값 손실이 발생할 수 있다.

정수



실수



## 자동형변환의 규칙

- 바이트 크기가 작은 자료형에서 큰 자료형으로 대입하면 형변환이 자동으로 이루어진다.
- 덜 정밀한 자료형에서 더 정밀한 자료형으로 대입하면 형변환이 자동으로 이루어진다.

자동형변환

큰 범위 타입 = 작은 범위 타입

`int = byte`

자동형변환

더 정밀한 범위 타입 = 덜 정밀한 범위 타입

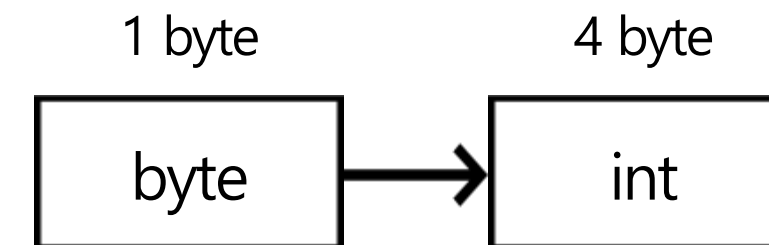
`float = int`

## 자동형변환과 강제형변환의 차이점

- 자동형변환 예시를 보면 byte형은 1바이트이고 int형은 4바이트이므로 자료손실없이 값이 옮겨진다.
- 강제형변환 예시를 보면 값 1000은 byte형 범위를 넘기 때문에 값이 손실되었다.

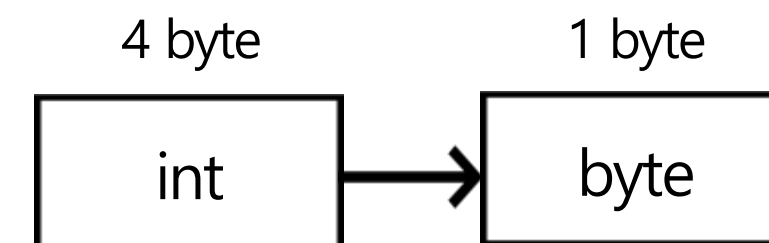
```
byte b = 10;
```

```
int i = b;    → 출력결과: 10
```



```
int i = 1000;
```

```
byte b = (byte) i;    → 출력결과: -24
```



부호비트가 잘려서 음수로 변함