

JAVA

1. 변수(VARIABLE)

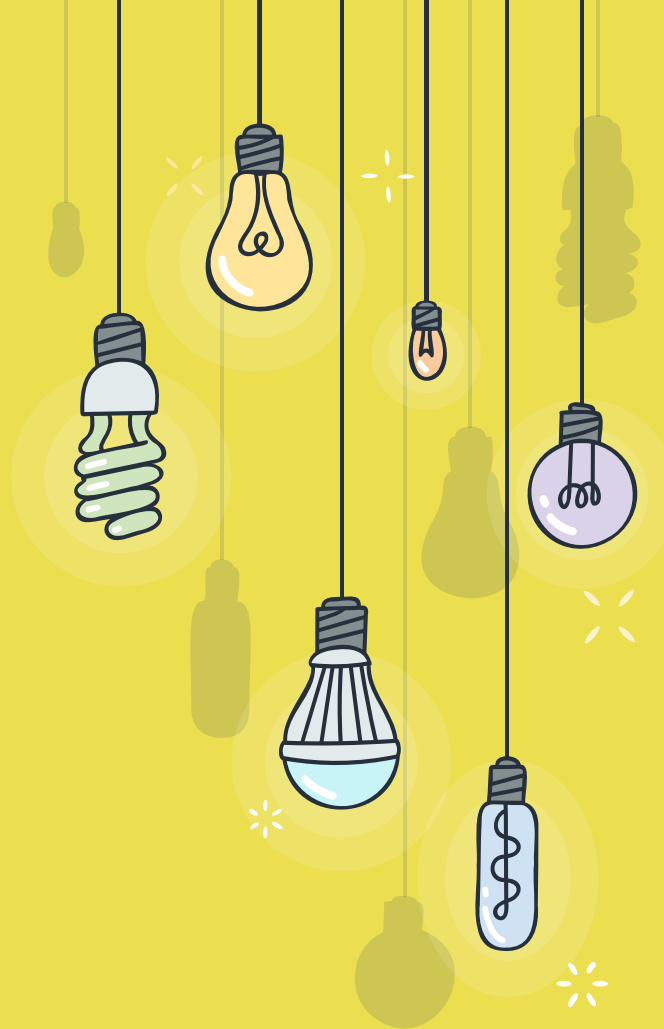
2. SCANNER

3. CASTING



1

변수 (VARIABLE)



# \* 변수

- + 하나의 값을 저장할 수 있는 메모리 공간
- + 기본(primitive) 자료형

값의 종류	기본 타입	메모리 사용 크기		저장되는 값의 범위
정수	byte	1 byte	8 bit	$2^7 \sim 2^7 - 1$ (-128~127)
	char	2 byte	16 bit	$0 \sim 2^{16} - 1$ (유니코드: \u0000~\uFFFF, 0~65535)
	short	2 byte	16 bit	$-2^{15} \sim 2^{15} - 1$ (-32,768~32,767)
	int	4 byte	32 bit	$-2^{31} \sim 2^{31} - 1$ (-2,147,483,648~2,147,483,647)
	long	8 byte	64 bit	$-2^{63} \sim 2^{63} - 1$
실수	float	4 byte	32 bit	$(+/-)1.4E-45 \sim (+/-)3.4E38$
	double	8 byte	64 bit	$(+/-)4.9E-324 \sim (+/-)1.7E308$
논리	boolean	1 byte	8 bit	true, false

# \* 기본자료형

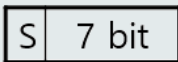
1 bit



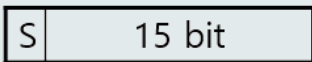
8 bit = 1 byte



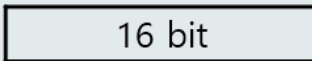
byte  $-2^7 \sim 2^7-1$



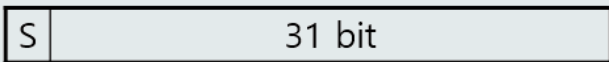
short  $-2^{15} \sim 2^{15}-1$



char  $0 \sim 2^{16}-1$



int  $-2^{31} \sim 2^{31}-1$



long  $-2^{63} \sim 2^{63}-1$



float 32 bit = 4 byte

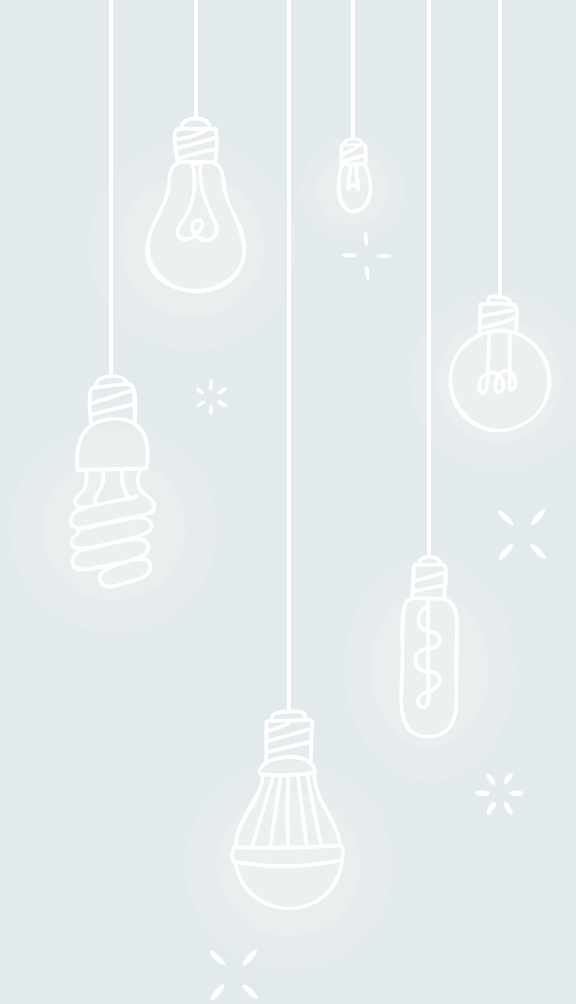


double 64 bit = 8 byte



# \* 참조 자료형

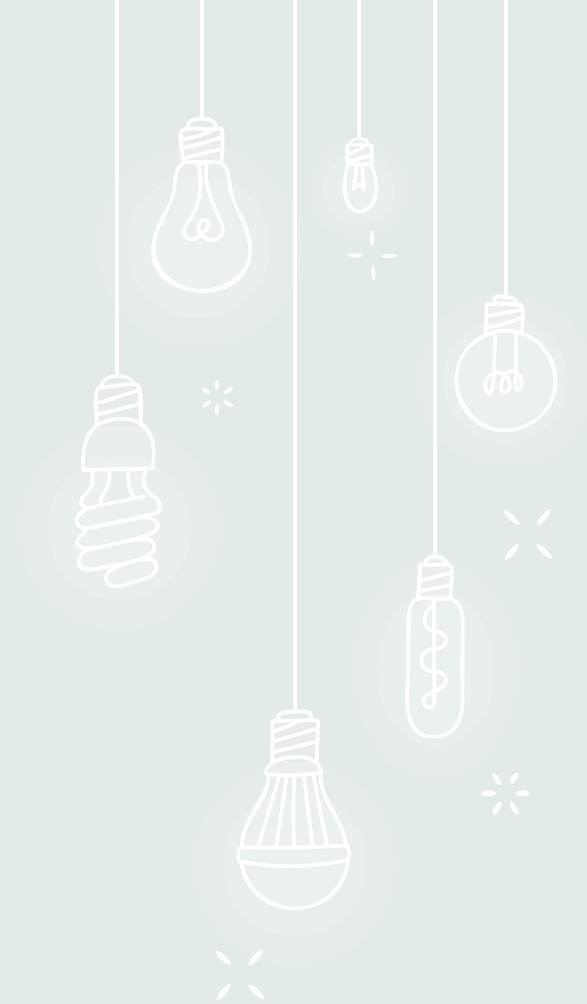
- + 참조형(Reference type) 자료형
  - × 기본형을 제외한 나머지(String, System 등)
  - × 객체의 주소 저장(4 byte)



# \* 변수 선언

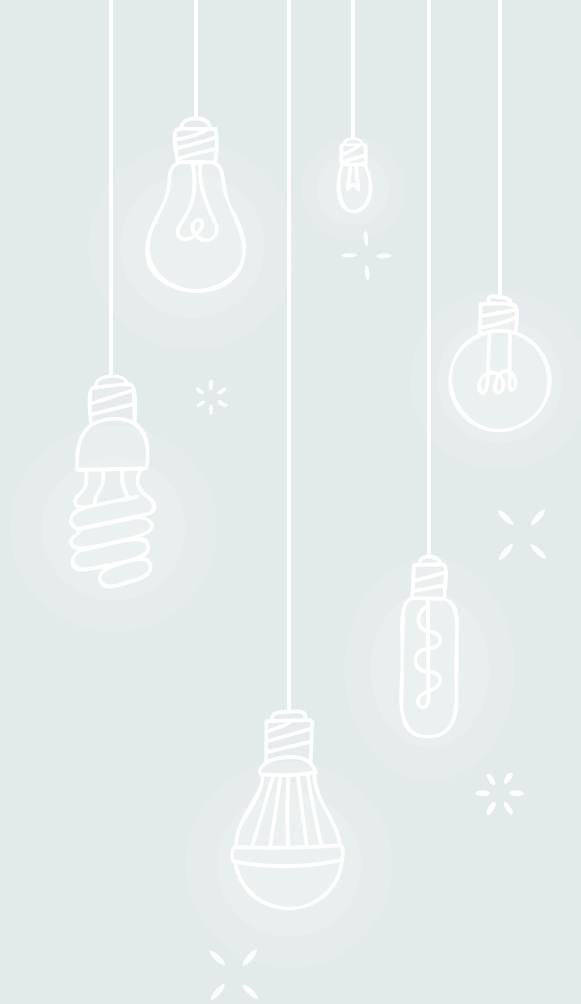
자료형 변수명;  
자료형 변수명 = 값;

```
int num;  
num = 10;  
int num = 10;
```



# \* 변수 명명규칙

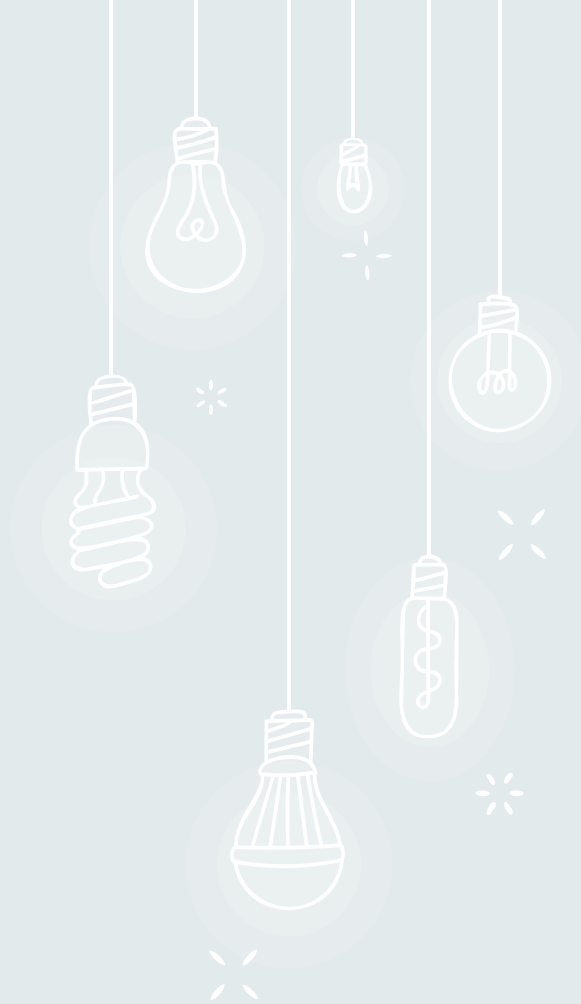
- + 대소문자를 구분한다
- + 예약어(Reserved word)를 사용해서는 안 된다.
- + 숫자로 시작해서는 안 된다
  - × 뒤에 붙는것은 가능
- + 특수문자는 '\_'와 '\$'만을 허용한다.
- + 길이 제한이 없다





# \* 변수, 상수, 리터럴

- + 변수(variable)
  - × 하나의 값을 저장하기 위한 공간
- + 상수(constant)
  - × 한 번만 값을 저장할 수 있는 공간
- + 리터럴(literal)
  - × 그 자체로 값을 의미하는 것



## \* 명명규칙-권장사항

- + 클래스 이름의 첫 글자는 대문자로 한다
- + 카멜 표기법 사용
  - × 맨 첫 글자를 제외한 각 합성어의 첫 글자만 대문자로 표기
  - × camelCase(일반 변수), isCamelCase(boolean 타입)
- + 상수의 이름은 대문자로 한다
  - × 단어는 '\_'로 구분
  - × PI, MAX, MIN\_SCORE

## \* 형식화된 출력

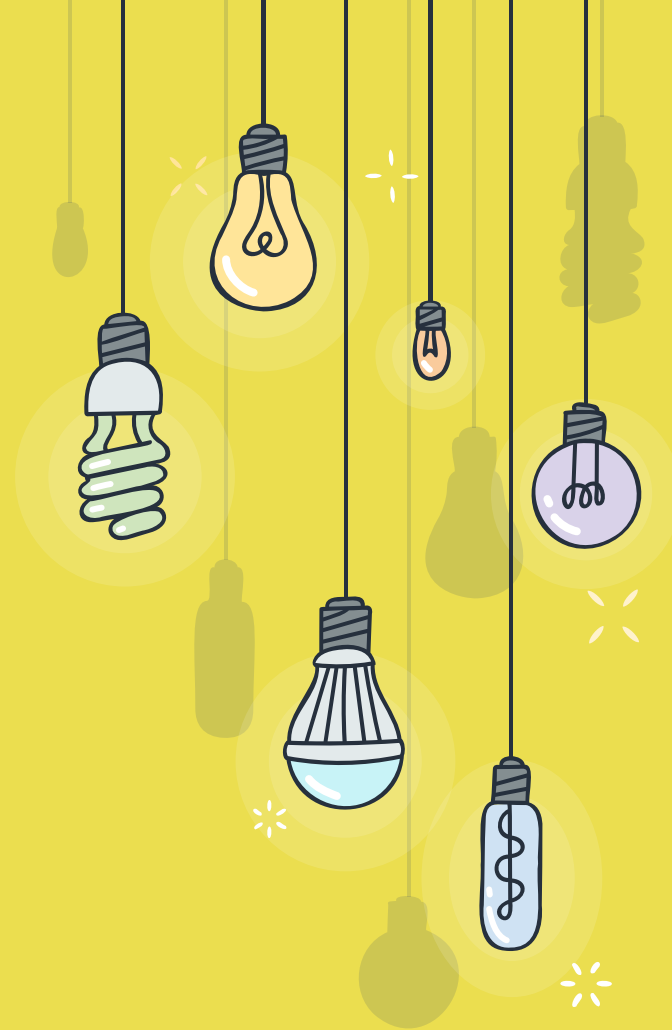
```
System.out.printf("%d", 5)
```

```
System.out.format("%d", 5)
```

지시자	설명
%b	boolean형식으로 출력
%d	10진 정수 형식으로 출력
%o	8진 정수 형식으로 출력
%x, %X	16진 정수 형식으로 출력
%f	실수형(float)으로 출력
%c	문자로 출력
%s	문자열로 출력

2

SCANNER



# \* SCANNER

- + 화면에서 입력 받기
- + System.in
  - × 키보드 장치를 직접 제어하고 키를 입력을 받는 java의 표준 입출력 스트림
  - × 키 값을 바이트 정보로 리턴
- + Scanner 클래스
  - × System.in으로 받은 키 값(바이트 정보)들을 문자, 정수, 실수, 불린, 문자열 등으로 변환
  - × java.util.Scanner 클래스

```
import java.util.*;  
Scanner scan = new Scanner(System.in);  
String str1 = scan.next();
```

## \* SCANNER 클래스의 주요 메서드

지시자	설명
String next()	단어 단위로 문자열 리턴
String nextLine()	띄어쓰기를 포함하여 한 라인을 읽음.
int nextInt()	int 를 읽음.
long nextLong()	long 을 읽음
float nextFloat()	float 을 읽음
double nextDouble()	double 을 읽음
boolean nextBoolean()	boolean 을 읽음

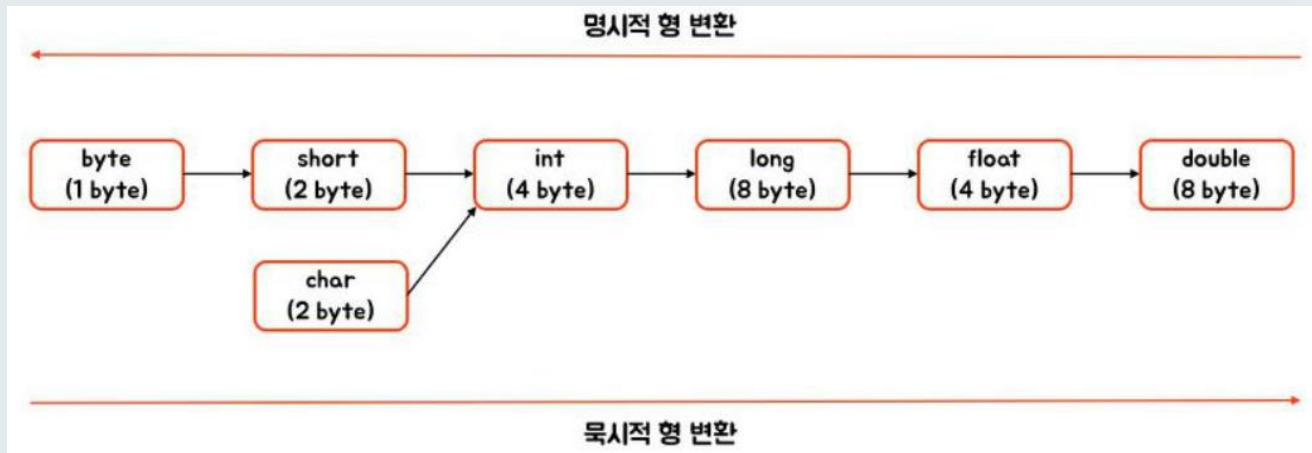
3

CASTING



# \* 타입 변환

- + 값의 타입을 다른 타입으로 변환하는 것
- + boolean을 제외한 7개의 기본형은 서로 타입변환 가능
- + 종류
  - × 자동(묵시적) 타입 변환: Promotion
  - × 강제(명시적) 타입 변환: Casting



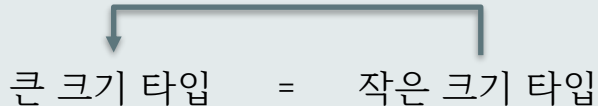


# \* 타입 변환

+ 자동(묵시적) 타입 변환: Promotion

× 작은 타입을 큰 타입으로 자동 타입 변환

자동 타입 변환



byte(1) < short, char (2) < int(4) < long(8) < float(4) < double(8)

× 예외: char 타입보다 허용 범위가 작은 byte 타입은 char 타입으로 자동 변환될 수 없음

# \* 타입 변환

## + 강제(명시적) 타입 변환: Casting

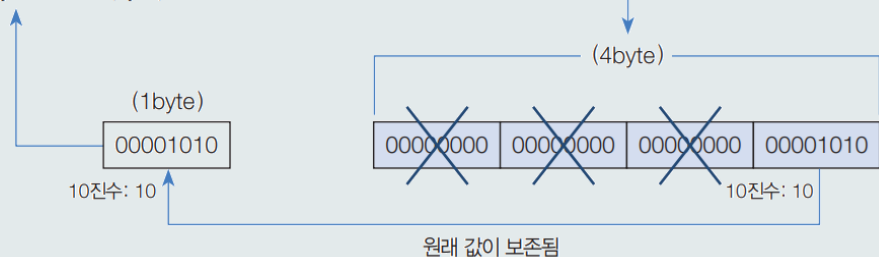
- × 큰 타입을 작은 타입 단위로 타입 변환
- × 끝의 한 부분만 작은 타입으로 강제적 변환

강제 타입 변환

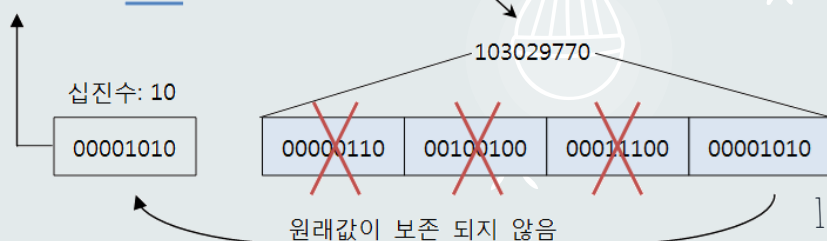
작은 크기 타입 = (작은크기타입)큰 크기 타입

## × int => byte로 강제 타입 변환

```
int intValue = 10;  
byte byteValue = (byte) intValue;
```



```
int intValue = 103029770;  
byte byteValue = (byte) intValue;
```



# \* 연산식에서 자동타입 변환

- + 연산식 두 피연산자 중 크기가 큰 타입으로 자동 타입 변환

```
int intValue = 10;
```

```
double doubleValue = 3.14;
```

double 로 자동 타입 변환

```
double result = intValue + doubleValue;
```

# \* 연산식에서 자동타입 변환

## + int 이하의 타입 연산

int result =  $\left[ \begin{array}{c} \text{byte type} \\ \text{char type} \\ \text{short type} \\ \text{int type} \end{array} \right]$  연산자(+, -, \*, /, %)

$\left[ \begin{array}{c} \text{byte type} \\ \text{char type} \\ \text{short type} \\ \text{int type} \end{array} \right]$

## + long타입 연산

long result =  $\left[ \begin{array}{c} \text{long type} \end{array} \right]$  연산자(+, -, \*, /, %)

$\left[ \begin{array}{c} \text{byte type} \\ \text{char type} \\ \text{short type} \\ \text{int type} \end{array} \right]$

# \* 연산식에서 자동타입 변환

## + 실수 타입 연산

double result =  $\left[ \begin{array}{c} \text{byte type} \\ \text{char type} \\ \text{short type} \\ \text{int type} \end{array} \right]$

연산자(+, -, \*, /, %)

$\left[ \begin{array}{c} \text{byte type} \\ \text{char type} \\ \text{short type} \\ \text{int type} \\ \text{float type} \\ \text{double type} \end{array} \right]$

## + float 연산 : 모두 float일 경우에

float result =  $\left[ \text{float type} \right]$  연산자(+, -, \*, /, %)  $\left[ \text{float type} \right]$

# THANKS!

+ Any questions?

