

# Java 入門

第 2 回

変数 (Variable)

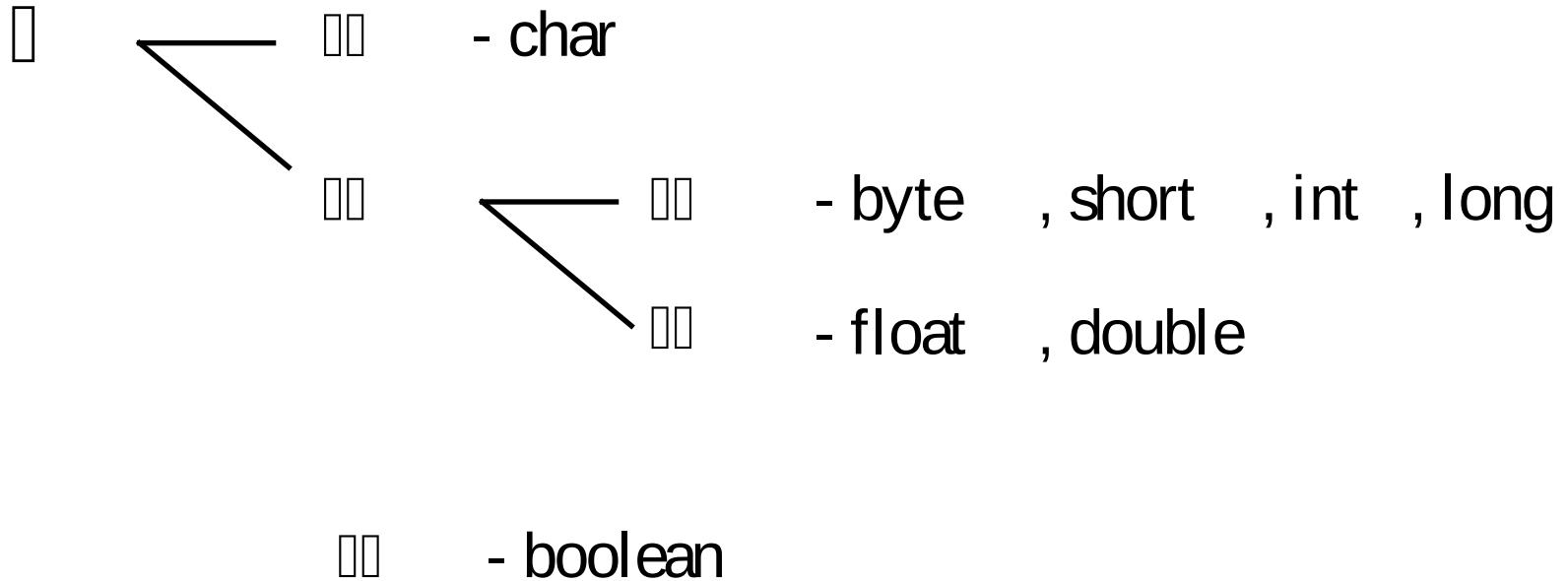
1. 변수 (Variable) 란 ?
2. 변수의 종류 (Type)
3. 변수의 선언
4. 네이밍 컨벤션 (Naming Convention)
5. 지역 변수, 전역 변수, 클래스 변수
6. 변수의 범위
7. 변수의 초기화
8. 변수의 생명주기
9. 런타임 오류 (Overflow)
10. 캐스팅 (Casting)

# 1. 変数 (Variable) とは？

変数とは？

変数とは、変化する値を格納するための場所のこと。

## 2. 데이터 타입 (Data type)



## 2. データ型 (Data type)

### ▶ 基本型 (Primitive type)

- 8 種類 (boolean, char, byte, short, int, long, float, double)
- 値を直接保持

### ▶ 参照型 (Reference type)

- 文字列 参照型 (String, System.out)
- 参照型参照型 (4 byte, 0x00000000~0xffffffff)

## 基本型 (Primitive type)

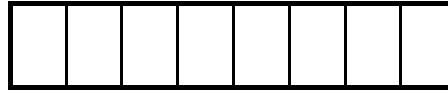
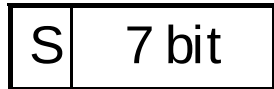
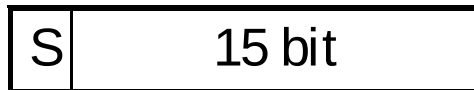
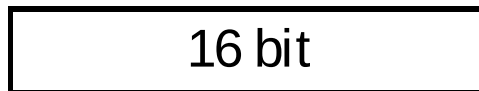
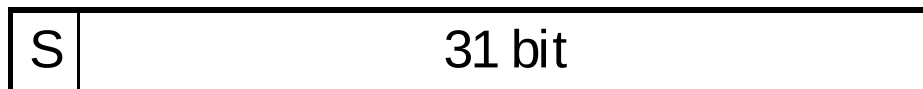
- ▶ `boolean` – `true` と `false` の 2 つの値を持つ。変数に `true` または `false` を代入する。
- ▶ `char` – 1文字の文字を表す。変数に 1文字の文字を代入する。
- ▶ `int` / `long` – 整数を表す。変数に整数を代入する。  
`byte` – 8ビットの整数を表す。  
`short` – 16ビットの整数を表す。
- ▶ `float` / `double` – 浮点小数を表す。変数に浮点小数を代入する。

変数の型	1	2	4	8
<code>boolean</code>	<code>boolean</code>			
<code>char</code>		<code>char</code>		
<code>byte</code>	<code>byte</code>	<code>short</code>		<code>long</code>
<code>float</code>			<code>float</code>	

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  $1+8+23=32$  bit = 4 bytedouble  $1+11+52=64$  bit = 8 byte

### 3. 変数の宣言と初期化

変数 宣言 ;

```
int score ;
```

```
score = 100;
```

```
int score = 100;
```

```
String str = new String("abc");
```

```
str = null;
```



## 4. 命名規則 (Naming convention)

1. 変数名は小文字で始める。
  - True は true の大文字で書くべきではない。
2. 予約語 (Reserved word) は変数名にしない。
  - true は変数名にしない、True は変数名にしない。
3. 変数名は小文字で始める。
  - top10 は変数名、7up は変数名にしない。
4. 変数名は '\_' や '\$' を使わない。
  - Sharp は変数名、S#arp は変数名にしない。

## 4. 상수 - 변수

1. 상수 상수 는 변수 상수 상수 상수 .  
- 상수 상수 상수 는 상수 상수 상수 상수 .
2. 상수 상수 상수 상수 는 상수 상수 상수 상수 .  
- lastIndexOf, StringBuffer
3. 상수 상수 상수 상수 상수 . 상수 ' \_ ' 는 상수 상수 상수 .  
- PI, MAX\_NUMBER

## 5. 변수, 상수, 리터럴

- ▶ 변수 (variable) – 값을 저장할 수 있는 공간
- ▶ 상수 (constant) – 값이 변하지 않는 변수
- ▶ 리터럴 (literal) – 값을 나타내는 문자열

```
int score = 100;
```

```
score = 200;
```

- ```
char ch = 'A';
```

```
String str = "abc";
```

```
final int MAX = 100;
```

```
MAX = 200; // 오류
```

## 6. 変数の宣言

```
boolean power = true;
```

```
char ch = 'A';
```

```
char ch = '\u0041';
```

```
char tab = '\t';
```

```
byte b = 127;
```

```
short s = 32767;
```

```
int i = 100;
```

```
int oct = 0100;
```

```
int hex = 0x100;
```

```
long l = 1000000000000L;
```

```
float f = 3.14f
```

```
double d = 3.14d
```

```
float f = 100f;
```

```
10.    —————> 10.0
```

```
.10    —————> 0.10
```

```
10f    —————> 10.0f
```

```
3.14e3f —————> 3140.0f
```

```
1e1    —————> 10.0
```

# 7. 変数の初期化

変数 名前 : 変数 初期値 変数 初期値

\* 初期化 変数 変数 初期値 初期値 初期値 .

| 変数      | 初期値         |
|---------|-------------|
| boolean | false       |
| char    | '\u0000'    |
| byte    | 0           |
| short   | 0           |
| int     | 0           |
| long    | 0L          |
| float   | 0.0f        |
| double  | 0.0d 変数 0.0 |
| 変数 初期値  | null        |

```
boolean isGood = false;
```

```
char grade = ' '; // 変数
```

```
byte b = 0;
```

```
short s = 0;
```

```
int i = 0;
```

```
long l = 0; // 0L 変数
```

```
float f = 0; // 0.0f 変数
```

```
double d = 0; // 0.0 変数
```

```
String s1 = null;
```

```
String s2 = ""; // 変数
```

## 8. 文字列の連結

```
char ch = 'A';
```

```
char ch = 'AB'; // 文字列
```

```
String s1 = "AB";
```

```
char ch = "; // 文字列
```

```
String s1 = "";
```

```
String s1 = "A" + "B"; // "AB"
```

```
"" + 7 → "" + "7" → "7"
```

```
""+7+7 → "7"+7 → "7"+"7" → "77"
```

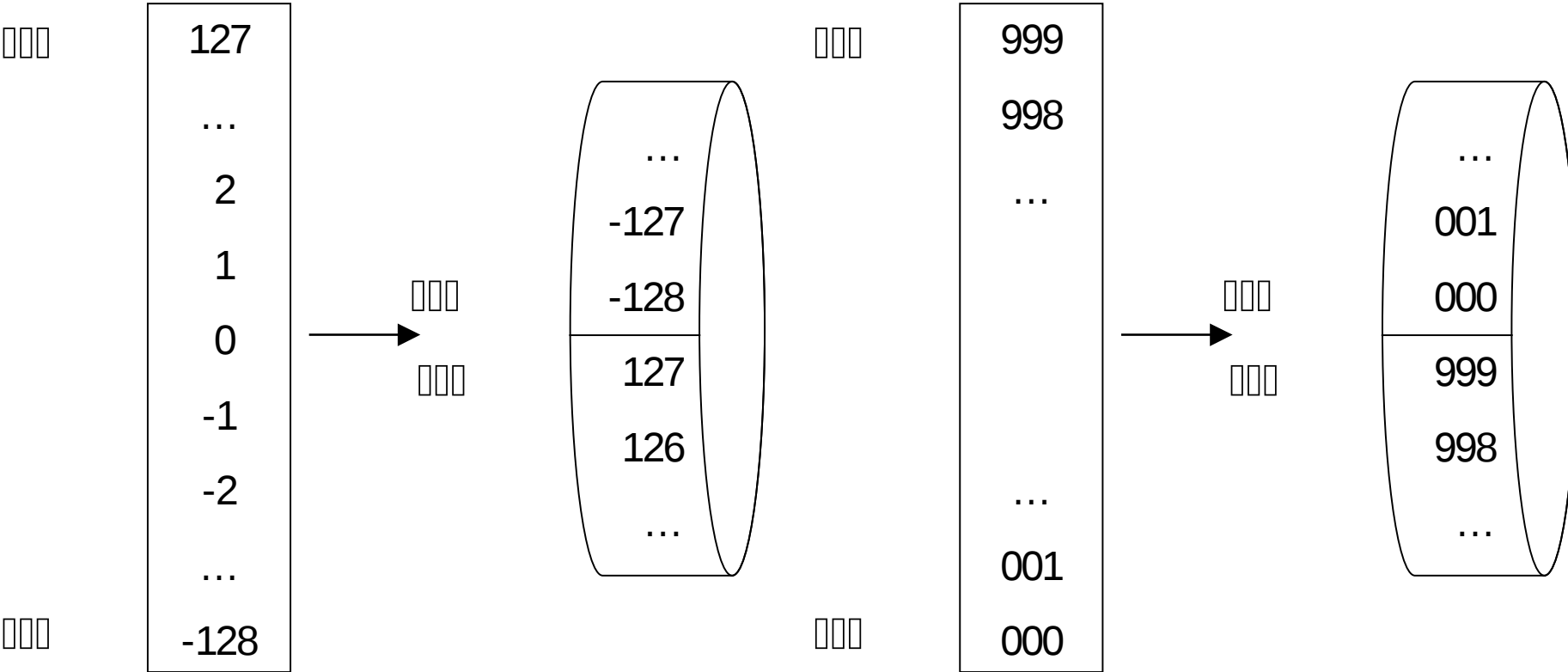
```
7+7+"" → 14+"" → "14"+"" → "14"
```

```
文字列 + any type 文字列 →
```

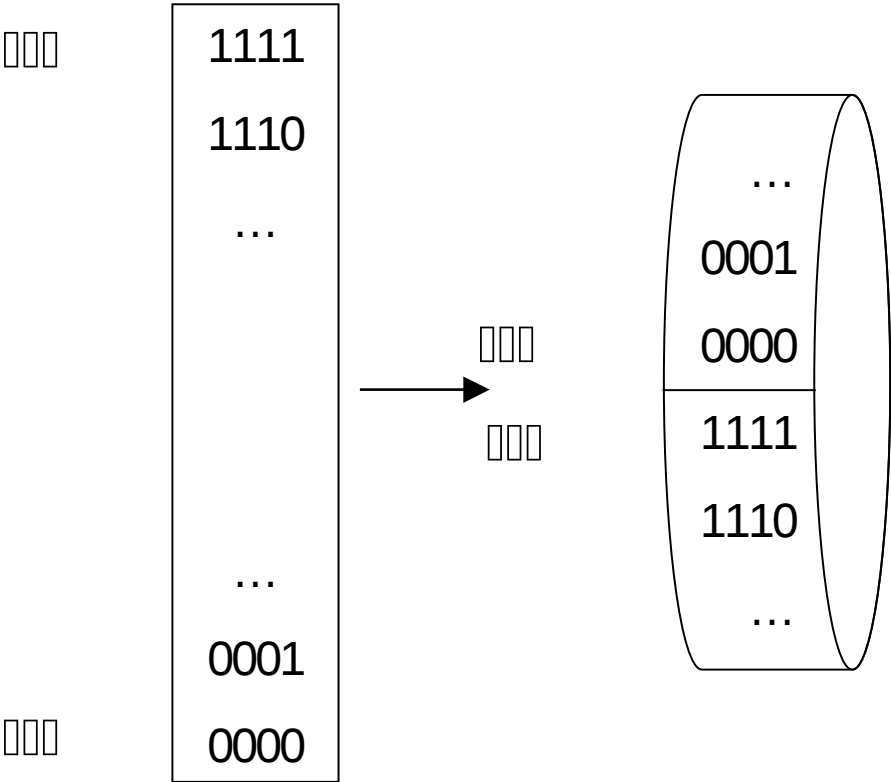
```
any type + 文字列 文字列 →
```

# 9. 変数 溢り (Overflow)

```
byte b = 127;    byte b = 128; // 変数
b = b + 1; // b が 変数 1 変数
```



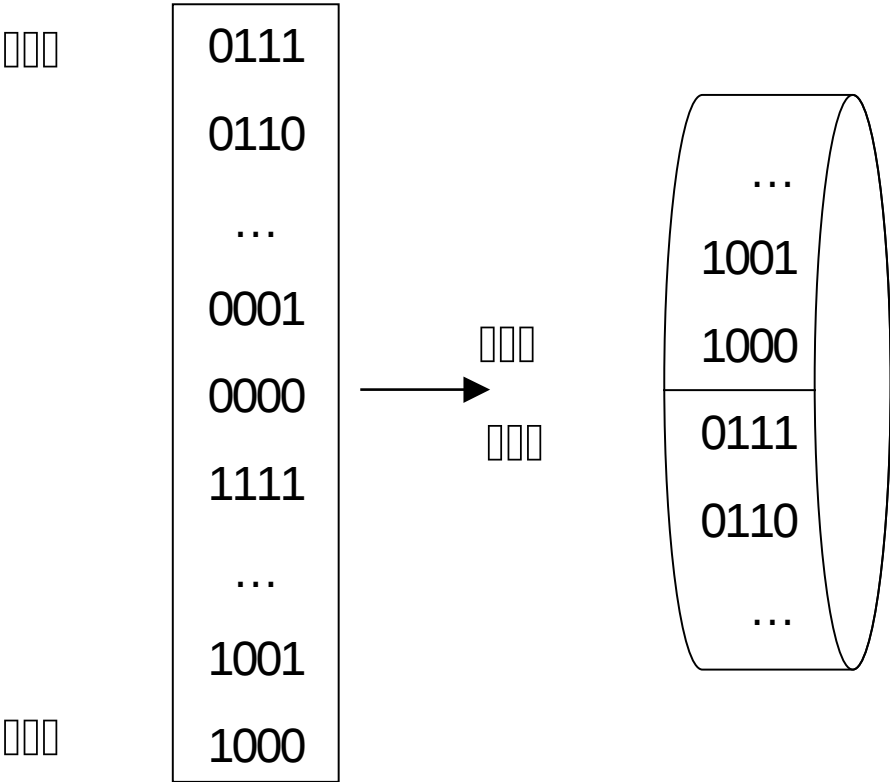
1. 変数 1 変数



1 変数 2 変数



2. 変数 2 変数



1 変数 2 変数





## 10. 형변환 (Casting)

왜 필요? ?

- 다른 타입의 변수를 같은 타입의 변수로 변환.
- boolean 타입은 7개의 기본 타입 중 유일하게 형변환이 불가능.

```
float f = 1.6f;
```

```
int i = (int)f;
```

| 원본 타입       | 변환 방법     | 결과    |
|-------------|-----------|-------|
| int → char  | (char)65  | 'A'   |
| char → int  | (int)'A'  | 65    |
| float → int | (int)1.6f | 1     |
| int → float | (float)10 | 10.0f |

## 2. int $\rightarrow$ byte

```
int i2 = 300;
```

```
byte b2 = (byte)i2; // 0000
```

```
graph LR; byte[byte] --> short[short]; short --> int[int]; int --> long[long]; long --> float[float]; float --> double[double]; char[char] --> int
```

1 byte      2 byte      4 byte      8 byte      4 byte      8 byte

byte → short → int → long → float → double

char → int

