
제2장

변수(Variable)

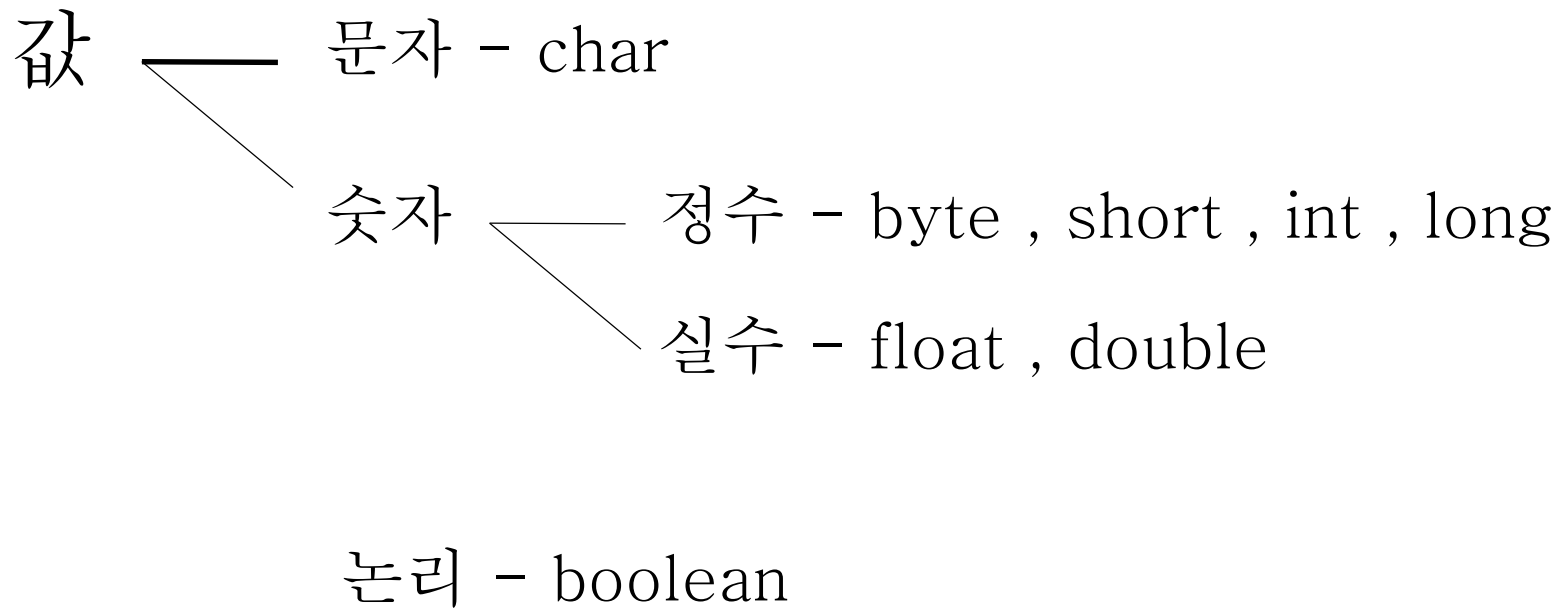
1. 변수(Variable)란?
2. 변수의 타입(Type)
3. 변수의 선언방법
4. 명명규칙(Naming Convention)
5. 변수, 상수, 리터럴
6. 리터럴과 접미사
7. 변수의 기본값과 초기화
8. 문자와 문자열
9. 정수의 오버플로우(Overflow)
10. 형변환(Casting)

1. 변수(Variable)란?

변하는 수?

하나의 값을 저장할 수 있는 기억공간

2. 변수의 타입(Data type)



2. 변수의 타입(Data type)

▶ 기본형(Primitive type)

- 8개 (boolean, char, byte, short, int, long, float, double)
- 실제 값을 저장

▶ 참조형(Reference type)

- 기본형을 제외한 나머지(String, System 등)
- 객체의 주소를 저장(4 byte, 0x00000000~0xffffffff)

기본형(Primitive type)

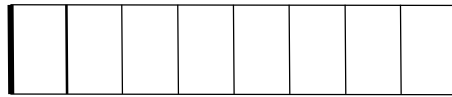
- ▶ 논리형 - true와 false중 하나를 값으로 갖으며, 조건식과 논리적 계산에 사용된다.
- ▶ 문자형 - 문자를 저장하는데 사용되며, 변수 당 하나의 문자만을 저장할 수 있다.
- ▶ 정수형 - 정수 값을 저장하는데 사용된다. 주로 사용하는 것은 int와 long이며, byte는 이진데이터를 다루는데 사용되며, short은 c언어와의 호환을 위해 추가되었다.
- ▶ 실수형 - 실수 값을 저장하는데 사용된다. float와 double이 있다.

크기 종류	1	2	48	
논리형	boolean			
문자형		char		
정수형	byte	short	int	long
실수형			float	double

1 bit



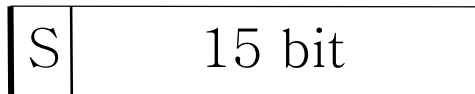
8 bit = 1 byte



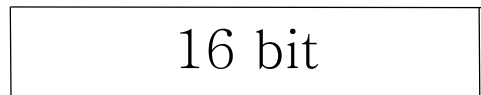
byte $-2^7 \sim 2^7-1$



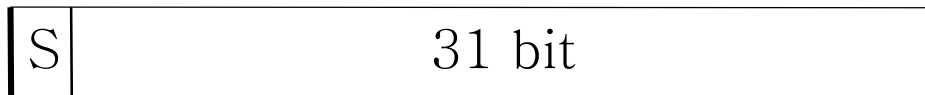
short $-2^{15} \sim 2^{15}-1$



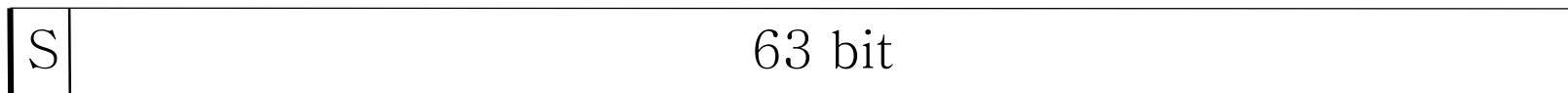
char $0 \sim 2^{16}-1$



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



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



float $1 + 8 + 23 = 32$ bit = 4 byte



double $1 + 11 + 52 = 64$ bit = 8 byte



3. 변수의 선언방법

타입 변수명;

```
int score ;
```

```
score = 100;
```

```
int score = 100;
```

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

```
    str = null;
```


4. 명명규칙(Naming convention)

1. 대소문자가 구분되며 길이에 제한이 없다.
 - True와 true는 서로 다른 것으로 간주된다.
2. 예약어(Reserved word)를 사용해서는 안 된다.
 - true는 예약어라 사용할 수 없지만, True는 가능하다.
3. 숫자로 시작해서는 안 된다.
 - top10은 허용하지만, 7up은 허용되지 않는다.
4. 특수문자는 '_'와 '\$'만을 허용한다.
 - \$sharp은 허용되지만 S#arp는 허용되지 않는다.

4. 명명규칙 - 권장사항

1. 클래스 이름의 첫 글자는 항상 대문자로 한다.
 - 변수와 메서드 이름의 첫 글자는 항상 소문자로 한다.
2. 여러 단어 이름은 단어의 첫 글자를 대문자로 한다.
 - lastIndexOf, StringBuffer
3. 상수의 이름은 대문자로 한다. 단어는 '_'로 구분한다.
 - PI, MAX_NUMBER

5. 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

```
int score = 100;
```

```
score = 200;
```

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

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

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

```
MAX = 200; // 예러
```

6. 리터럴과 접미사

```
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. \longrightarrow 10.0

.10 \longrightarrow 0.10

10f \longrightarrow 10.0f

3.14e3f \longrightarrow 3140.0f

1e1 \longrightarrow 10.0

7. 변수의 기본값과 초기화

변수의 초기화 : 변수에 처음으로 값을 저장하는 것

* 지역변수는 사용되기 전에 반드시 초기화해주어야 한다.

자료형	기본값
boolean	false
char	'\u0000'
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d 또는 0.0
참조형 변수	null

```
boolean isGood = false;
```

```
char grade = ' '; // 공백
```

```
byte b = 0;
```

```
short s = 0;
```

```
int i = 0;
```

```
long l = 0; // 0L로 자동변환
```

```
float f = 0; // 0.0f로 자동변환
```

```
double d = 0; // 0.0로 자동변환
```

```
String s1 = null;
```

```
String s2 = ""; // 빈 문자열
```

8. 문자와 문자열

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

```
char ch = 'AB'; // 에러
```

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

```
char ch = "; // 에러
```

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

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

```
"" + 7 → "" + "7" → "7"
```

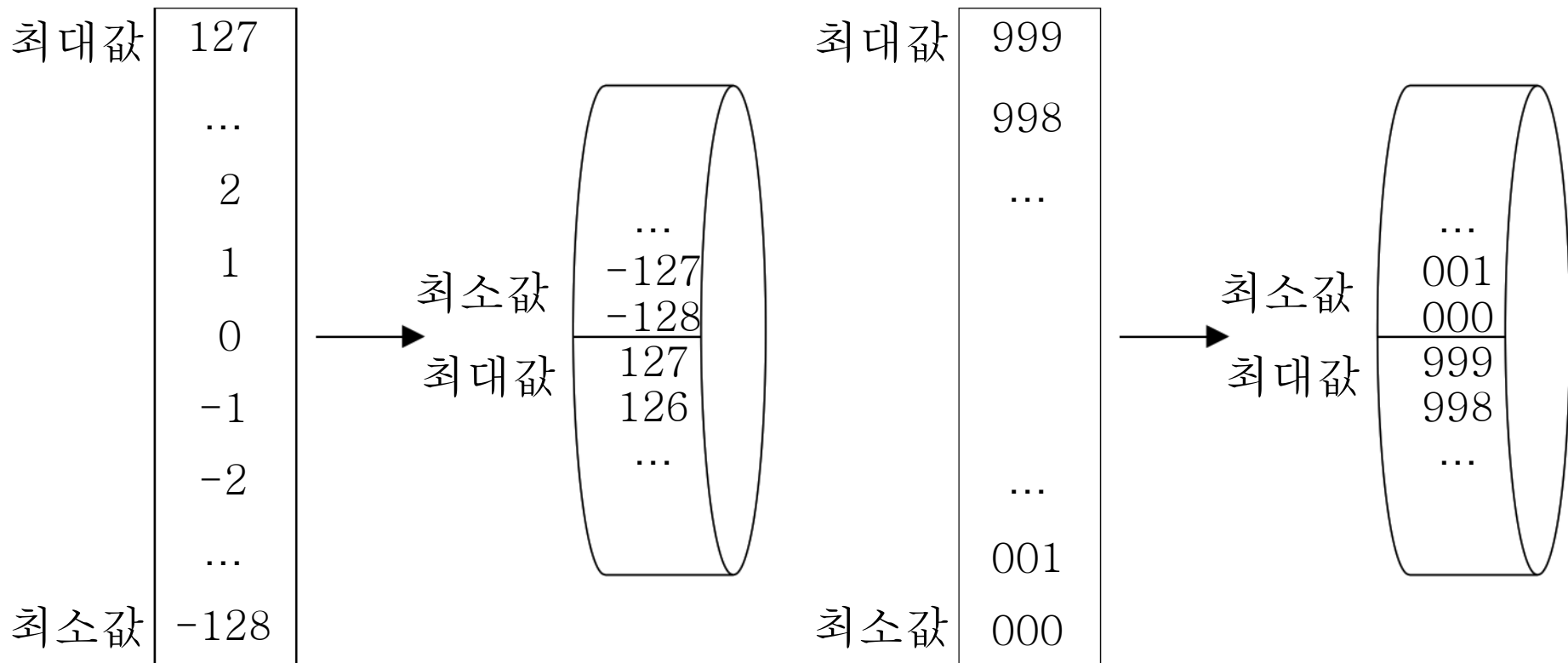
```
"" + 7 + 7 → "7" + 7 → "7" + "7" → "77"
7 + 7 + "" → 14 + "" → "14" + "" → "14"
```

문자열 + any type → 문자열

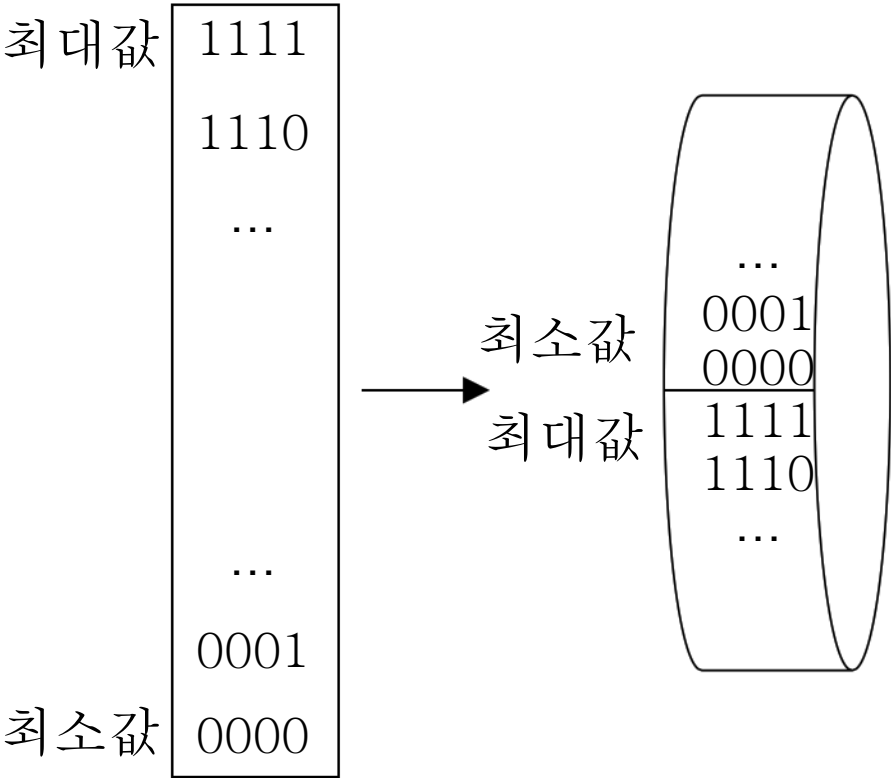
any type + 문자열 → 문자열

9. 정수의 오버플로우(Overflow)

```
byte b = 127;    byte b = 128; //에러
b = b + 1; // b에 저장된 값을 1증가
```



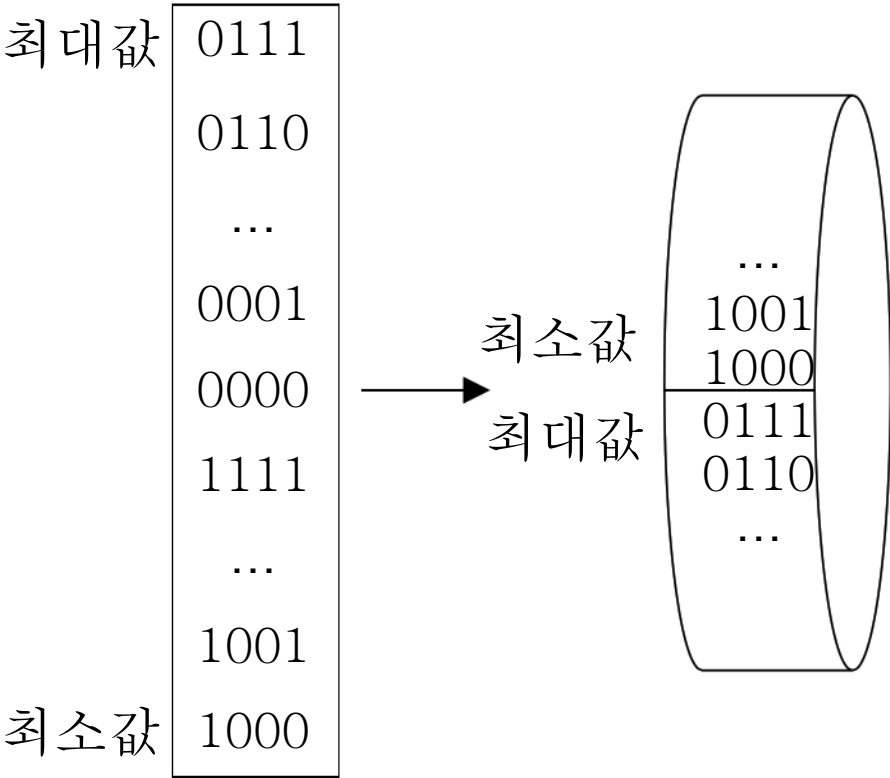
1. 부호가 없는 정수



1씩 증가하는 2진 카운터



2. 부호가 있는 정수



1씩 감소하는 2진 카운터



10. 형 변환(Casting)

형 변환이란?

- 값의 타입을 다른 타입으로 변환하는 것이다.
- boolean을 제외한 7개의 기본형은 서로 형변환이 가능하다.

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

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

변환	수식	결과
int → char	(char) 65	'A'
char → int	(int) 'A'	65
float → int	(int) 1.6f	1
int → float	(float) 10	10.0f

1. byte \rightarrow int

```
byte b = 10;
```

```
int i = (inbt;) // 생략가능
```

2. $\text{int} \rightarrow \text{byte}$

```
int i2 = 300;
```

```
byte b2 = (byte)i2; // 생략불가
```

변환	2진수	10진수	값손실
byte ↓ int	<div style="text-align: right;">0 0 0 0 1 0 1 0</div> <div>0 1 0 1 0</div>	10 10	없음
int ↓ byte	<div>0 1 0 0 1 0 1 1 0 0</div> <div style="text-align: right;">0 0 1 0 1 1 0 0</div>	300 44	있음

