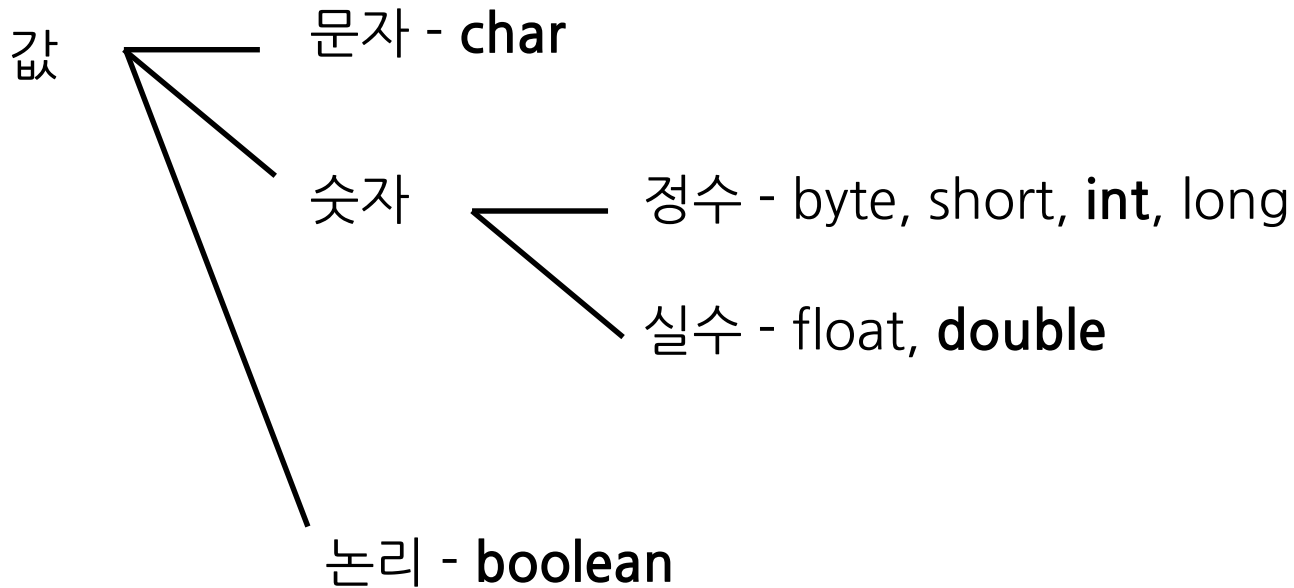


JAVA 프로그래밍

## 변수 & 자료형

# 변수의 종류



# 변수의 타입(Data type)

## ▶ 기본형(Primitive type)

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

## ▶ 참조형(Reference type)

- 기본형을 제외한 나머지(String, System 등)
- 객체의 위치를 알려주는 객체의 주소를 저장하는 변수

# 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 이진 데이터를 다루는데 사용되며, short은 c언어와의 호환을 위해 추가되었다.
- § **실수형** : 실수 값을 저장하는데 사용된다. float와 double이 있다.

크기 종류	1byte	2byte	4byte	8byte
논리형	boolean			
문자형		char		
정수형	byte	short	int	long
실수형			float	double

# 자료형의 크기와 범위

자료형	키워드	크기 (byte)	기본값	표현범위
논리	boolean	1	false	false, true
문자	char	2	₩u0000	0 ~ 65,535
정수	byte	1	0	-128~127
	short	2	0	-32,768 ~ 32,767
	int	4	0	-2,147,483,648 ~ 2,147,483,647
	long	8	0	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
실수	float	4	0.0f	-3.4E38 ~ +3.4E38
	double	8	0.0	-1.7E308 ~ +1.7E308

# 변수의 선언방법

- 선언 형식 : 자료형 타입 변수명(variable name)

- 예)

- `int score;`            `// 정수형 변수 선언`

- `score = 100;`        `// 변수 초기화`

- `int score = 100;` `// 정수형 변수 선언과 초기화`

- `String str = new String("abc");`    `// 문자열 변수 선언과 초기화`

# 명명 규칙(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.
  - \$harp은 허용되지만 S#arp는 허용되지 않는다.

# 명명규칙 - 권장사항

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



# 실습 - 변수 선언 규칙

〈파일 이름: Variables.java〉

```
public static void main(String[] args) {  
    int x=20;  
    int i = 123.456;  
    float f = 123.456f;  
    double d = 123.456;  
    int x = 2147483649;  
    char ch1 = 'A';  
    char ch2 = 'ABC';  
    String str = "ABC def";  
    int 1_a = 3;  
    System.out.println(x);  
    System.out.println(i);  
    System.out.println(f);  
    System.out.println(ch1);  
    System.out.println(str);  
}
```

# 변수, 상수, 리터럴

- ▶ 변수(variable) : 하나의 값을 저장하기 위한 공간, 변하는 수

```
int score = 100;           // score 변수 선언하면서 100으로 초기화
score = 200;               // score의 값을 200으로 변경
char ch = 'A';             // 문자형 ch 변수 선언과 초기화
String str = "abc";        // 문자열 str 변수 선언과 초기화
```

- ▶ 상수(constant) : 한 번만 값을 저장할 수 있는 공간, 변하지 않는 수

```
final int MAX = 100;       //상수 MAX 선언하면서 100으로 초기화
MAX = 200                  // 오류(상수의 값은 변경 안됨)
```

- ▶ 리터럴(literal) : 그 자체로 값을 의미하는 것 // 100, 200, A, abc 등

# 특수문자를 표현하는 방법

특수문자	리터럴
tab	\t
backspace	\b
new line(개행, 줄 바꿈)	\n
carriage return(커서를 현재 행의 맨 앞으로)	\r
역슬래쉬(\)	\\
작은 따옴표	'\''
큰 따옴표	'\"'
유니코드(16진수) 문자	\u유니코드 ( char a = '\u0041' )

# 문자와 문자열

〈파일 이름: CharString.java〉

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

```
char ch = 'AB';    // 오류
```

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

```
char ch = ' ';    // 빈칸(띄어쓰기)
```

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

〈문자열 결합〉

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

예) `" "+7 → ""+"7" → "7"`

`""+7+7 → "7"+7 → "7"+"7" → "77"`

`7+7+" " → 14+" " → "14"+" " → "14"`

〈결과〉

문자열 + any type → 문자열

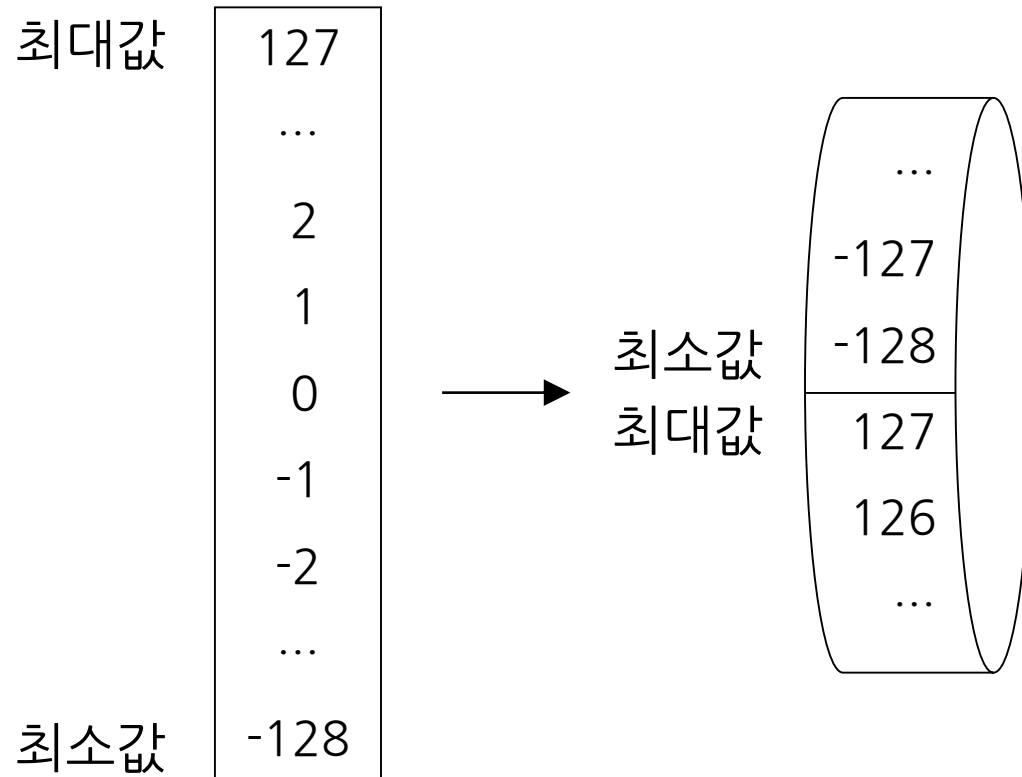
any type + 문자열 → 문자열

# 정수의 오버플로우(Overflow)

byte b = 127;

byte b = 128; // 오류

b = b + 1; // b에 저장된 값을 1증가, b의 값은 -128



# 실습 - 오버플로우와 언더플로우

〈파일 이름: Overflow.java〉

```
public static void main(String[] args)
{
    byte a = 127;
    byte b = -128;

    a = a+1;           // 컴파일 오류(오버플로우)
    b = b-1;           // 컴파일 오류(언더플로우)

    byte c = (byte) (a+1);
    System.out.println("오버플로우 발생 후 a의 값: " + c);

    byte d = (byte) (b-1);
    System.out.println("언더플로우 발생 후 b의 값: " + d);
}
```

# 형 변환(Type Casting)

형 변환이란?

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

예) float f = 1.6f;           // float형 변수 f에 1.6 저장  
    int i = (int)f;           // f에 저장된 1.6을 int형으로 변환하면  
                              // 자료형이 변경되어 변수 i에 저장

변 환	수 식	결 과
int → char	(char)65	'A'
char → int	(int)'A'	65
float → int	(int)1.6f	1
int → float	(float)10	10.0f

## 형 변환을 생략하는 경우

byte(1바이트) → int(4바이트)

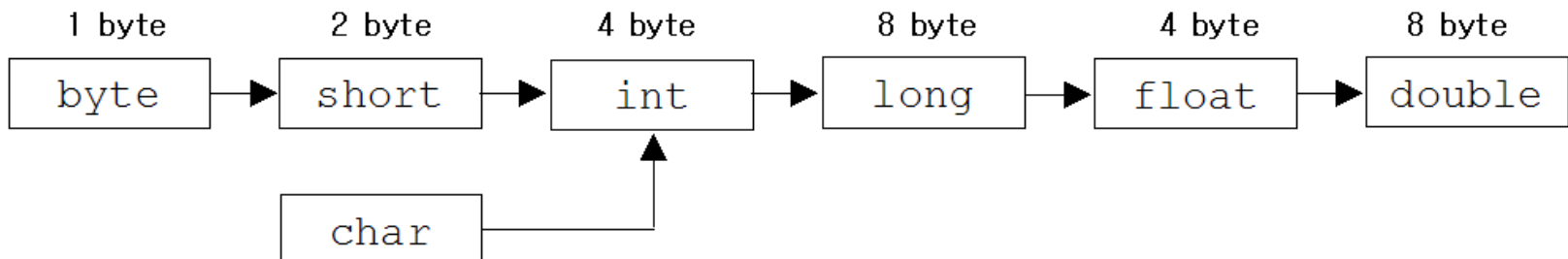
```
byte b = 10;
```

```
int i = b;    // (int)는 생략가능
```

**int(4바이트)  $\rightarrow$  byte(1바이트)**

```
int i2 = 300;
```

```
byte b2 = (byte)i2; // 생략불가
```

[illegible]



# 실습 - 형 변환

```
public class CastingTest{
    public static void main(String[] args) {
        long l = 12345L;  // int형 값을 long형 변수에 대입하면 자동 long형으로 변환
        int i = 35000;
        short s = 128;
        i = s;  // int형 변수에 short형 변수를 대입하면 자동 int형으로 변환
        System.out.printf("s값: %d \n", s);
        //float f = 23.7;          // 오류(float형 값 뒤에 'f'를 붙이지 않으면 double형으로 인식)
        float f = 23.7f;
        double d = 23.7;
        char c = 'A';          // char형을 int로 형변환하면 'A'의 아스키코드 값으로 출력
        System.out.printf("%c -> %d \n", c, (int)c);
        int a = 100;
        String s1 = "JavaClass";
        System.out.printf("s1+a = %s \n", s1+a); // String+다른 타입=String형으로 자동변환
        int b = 200;
        String s2 = "100";
        System.out.println("b+s2 = " + (b + Integer.parseInt(s2))); // 숫자 300 출력
    }
}
```

# Quiz

〈VariablesExample.java〉 각 변수들을 선언하고 초기화하여 아래와 같이 출력하시오.

[변수 선언]

```
num          // 번호
name         // 이름
kor, mat, eng // 국어, 수학, 영어
sum          // double형 총점
avg          // float형 평균
```

[출력 결과]

```
번호: 2014001
이름: 김나라
국어 점수: 100
수학 점수: 97
영어 점수: 99
총점: 296.0
평균: 98.666664
```

# Quiz

<Box.java>

변수 선언 : w, h, area, perimeter

면적  $\text{area} = w * h$

둘레  $\text{perimeter} = 2 * (w + h)$

[실행 결과]

사각형의 가로 : 10.25, 사각형의 세로 : 5.75

사각형의 면적 : 58.9375

사각형의 둘레 : 32.0

# Quiz

<Light.java>

변수 선언: speed=300000, distance=0

빛의 거리(distance) 계산식=speed\*365\*24\*60\*60

[실행 결과]

빛이 1년 동안 가는 거리: 9460800000000km

# Quiz

<Constant.java>

변수 선언: radius = 5, area

상수 선언: PI = 3.141592

원의 면적(area) 계산식 =  $PI * radius * radius$

[실행 결과]

원의 면적은 78.5398입니다.