

2장. 변수와 타입



Contents

1절. 변수

- 변수란?
- 변수의 선언
- 변수의 사용
- 변수의 사용 범위

2절. 데이터 타입

- 기본 타입
- 정수 타입
- 실수 타입
- 논리 타입

3절. 타입 변환

- 자동 타입 변환(Promotion)
- 강제 타입 변환(Casting)



1절. 변수

❖ 변수란?

변수란, 하나의 값을 저장할 수 있는 메모리 공간

❖ 변수의 선언

타입 변수이름

```
int age ;
```

```
double value ;
```



1절. 변수

❖ 변수 이름을 위한 명명 규칙(naming convention)

작성 규칙	예
첫번째 글자는 문자이거나 '\$', '_' 여야 하고 숫자로 시작할 수 없다. (필수)	가능: price, \$price, _companyName 안됨: 1v, @speed, \$#value
영어 대소문자가 구분된다. (필수)	firstname 과 firstName 은 다른 변수
첫문자는 영어 소문자로 시작하되, 다른 단어가 붙을 경우 첫자를 대문자로 한다. (관례)	maxSpeed, firstName, carBodyColor
문자 수(길이)의 제한은 없다.	
자바 예약어는 사용할 수 없다. (필수)	책 참조



1절. 변수

❖ 변수의 사용

■ 변수값 저장

```
int score;    //변수 선언  
score = 90;   //값저장
```

초기값은 변수를 선언함과 동시에 줄 수도 있다.

```
int score = 90;
```



1절. 변수

❖ 변수의 사용

■ 변수값 읽기

- 변수는 초기화가 되어야 읽기 가능
- 잘못된 코딩의 예

```
int value;                //변수 value 선언 (초기화 안됨)  
int result = value + 10;  //변수 value 값을 읽고 10 을 더한 결과값을 변수 result 에 저장
```

- 맞게 고친 후의 코드

```
int value = 30;           //변수 value 가 30 으로 초기화 됨  
int result = value + 10;  //변수 value 값을 읽고 10 을 더한 결과값(40)을 변수 result 에 저장
```



1절. 변수

❖ 리터럴(literal)

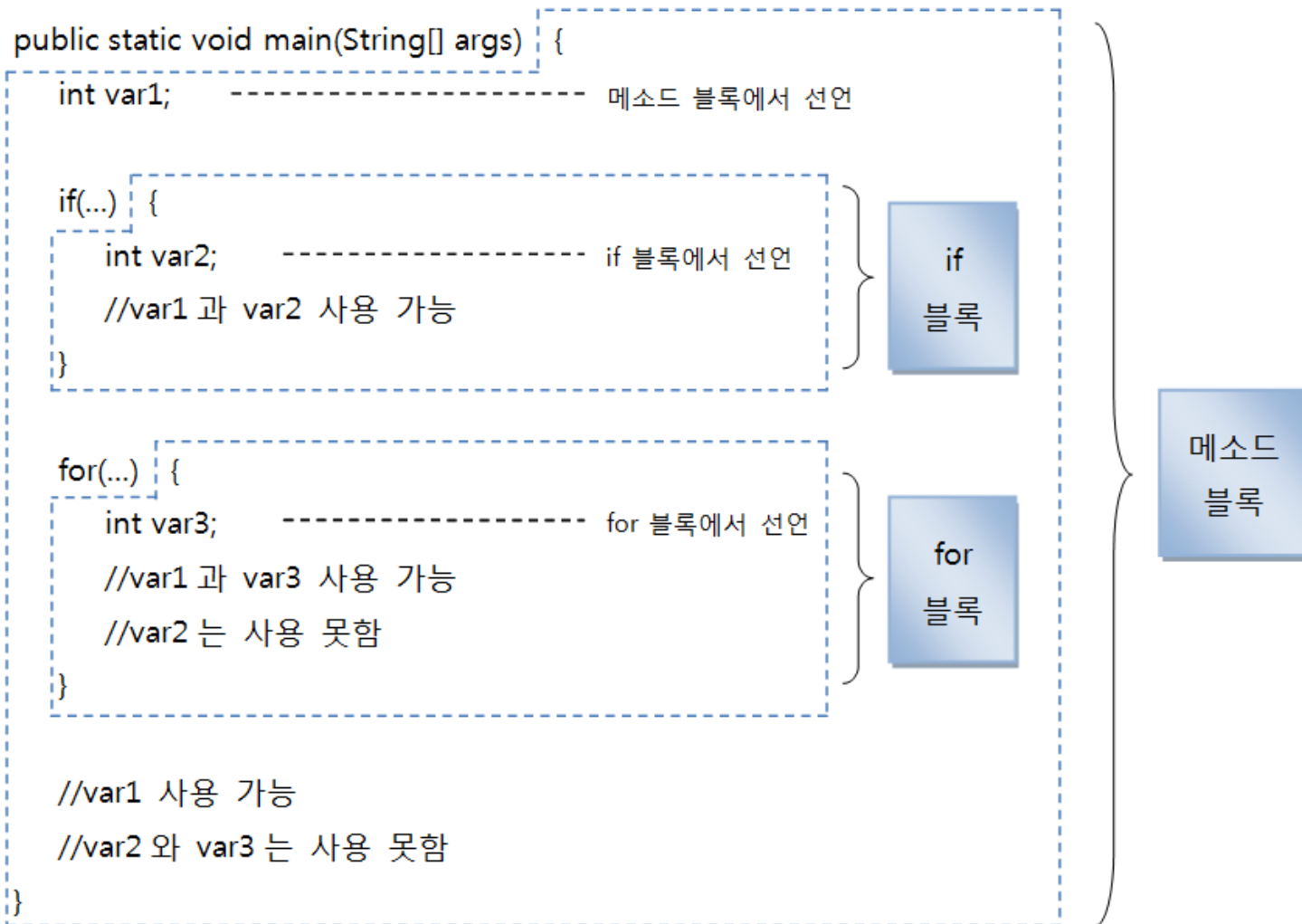
- 소스 코드 내에서 직접 입력된 변수의 초기값
- 소스 코드 내에서 익숙해지는 것이 point !
- 종류 : 정수 리터럴, 실수 리터럴, 문자 리터럴, 문자열 리터럴, 논리 리터럴



1절. 변수

❖ 변수의 사용 범위

- 변수는 중괄호 블록 {} 내에서 선언되고 사용



2절. 데이터 타입

❖ 기본(primitive) 타입

- 정수, 실수, 문자, 논리 리터럴을 직접 저장하는 타입
- 메모리의 최소 기억단위인 bit가 모여 byte 형성

1 byte = 8 bit

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

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 ~ (+/-)3.4E38
	double	8 byte	64 bit	(+/-)4.9E-324 ~ (+/-)1.7E308
논리	boolean	1 byte	8 bit	true, false



3절. 타입 변환

❖ 타입 변환

■ 데이터 타입을 다른 타입으로 변환하는 것

- byte ↔ int, int ↔ double

■ 종류

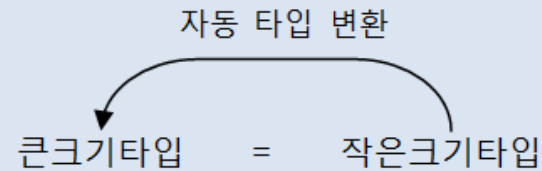
- 자동(묵시적) 타입 변환: Promotion
- 강제(명시적) 타입 변환: Casting



3절. 타입 변환

❖ 자동 타입 변환

- 프로그램 실행 도중 작은 타입은 큰 타입으로 자동 타입 변환 가능



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



3절. 타입 변환

❖ 강제 타입 변환

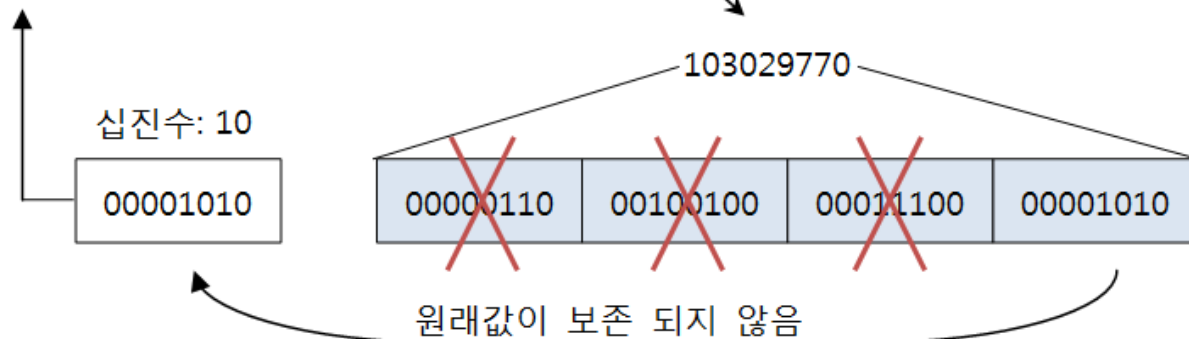
- 큰 타입을 작은 타입 단위로 쪼개기
- 끝의 한 부분만 작은 타입으로 강제적 변환

강제 타입 변환

작은크기타입 = (작은크기타입) 큰크기타입

• Ex) int 를 byte에 담기

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



3절. 타입 변환

❖ 연산식에서 자동 타입 변환

- 연산은 같은 타입의 피연산자(operand)간에만 수행
 - 서로 다른 타입의 피연산자는 같은 타입으로 변환
 - 두 피연산자 중 크기가 큰 타입으로 자동 변환

```
int intValue = 10;
```

```
double doubleValue = 5.5;
```

double 타입으로 자동 변환

```
double result = intValue + doubleValue; //result 에 15.5 가 저장
```

- Ex) int type으로 계산 결과를 얻고 싶다면?
 - Double type 변수를 먼저 int로 변환 후 계산

