

Chapter

# 02

## 변수와 타입



### 02-1. 변수

혼자 공부하는 자바(개정판) (신용권 저)

- 0. 시작하기 전에
- 1. 변수 선언
- 2. 값 저장
- 3. 변수 사용
- 4. 변수 사용 범위
- 5. 키워드로 끝내는 핵심 포인트



# 0. 시작하기 전에

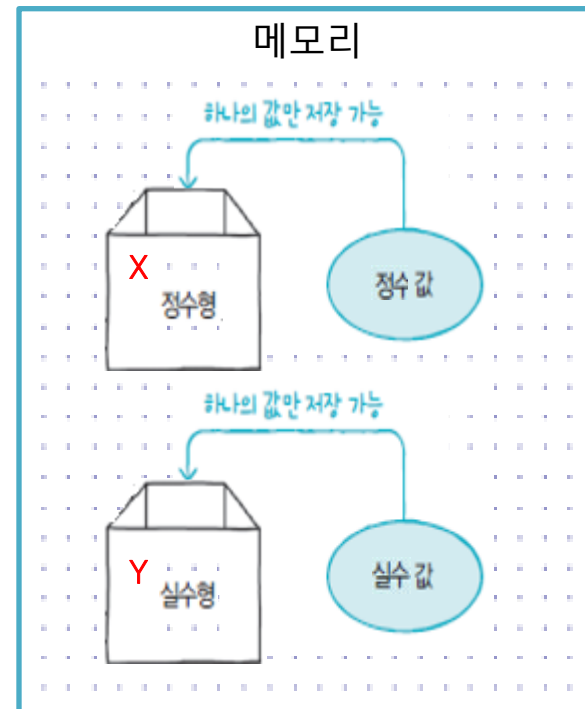
[핵심 키워드] : 변수, 변수 선언, 변수 사용, 변수 사용 범위

## [핵심 포인트]

- 컴퓨터 메모리(RAM)는 값을 저장할 수 있는 수많은 번지(주소)들로 구성되어 있다.
- 메모리의 어디에, 어떤 방식으로 저장할지 정해놓지 않으면 프로그램 개발이 무척 어렵게 된다.
- 프로그래밍 언어는 이 문제를 해결하기 위해 변수라는 개념을 사용한다.
- 변수의 역할 및 사용 방법에 대해 알아본다.

### ❖ 변수 (Variable)

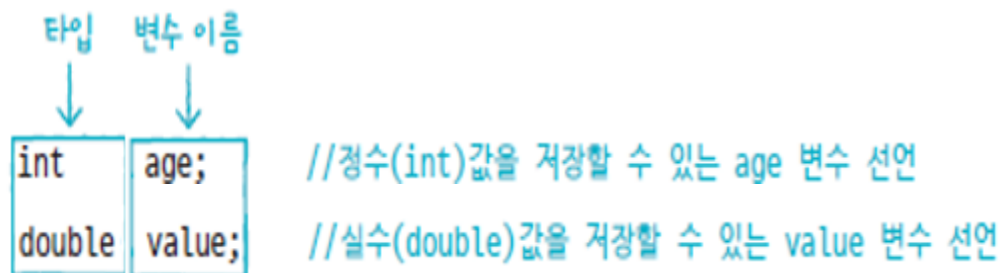
- 값을 저장할 수 있는 메모리의 특정 번지에 붙여진 이름
- 변수 통해 해당 메모리 번지에 하나의 값 저장하고 읽을 수 있음
- 변수는 정수, 실수 등 다양한 타입의 값을 저장할 수 있음



# 1. 변수 선언

## ❖ 변수 사용 위해서 변수 선언 필요

- 변수에 어떤 타입의 데이터 저장할 것인지, 변수 이름 무엇인지 결정



- 같은 타입의 변수는 콤마 이용해 한꺼번에 선언할 수 있음



# 1. 변수 선언

## ❖ 변수 이름

- 자바 언어에서 정한 명명 규칙 따라서 작성

작성 규칙	예
첫 번째 글자는 문자이거나 '\$', '_'이어야 하고 숫자로 시작할 수 없으며, 특수 문자가 포함되면 안됩니다(필수).	가능: price, \$price, _companyName 불가능: 1v, @speed, \$#value
영어 대소문자를 구분합니다(필수).	firstname과 firstName은 다른 변수
첫 문자는 영어 소문자로 시작하되, 다른 단어가 붙을 경우 첫 문자를 대문자로 합니다(관례).	maxSpeed, firstName, carBodyColor
문자 수(길이)의 제한은 없습니다.	
자바 예약어는 사용할 수 없습니다(필수).	다음 표 참조



# 1. 변수 선언

## ❖ 예약어

- 자바 언어에서 의미를 가지고 사용되는 단어
- 변수 이름으로 사용할 경우 컴파일 에러 발생

분류	예약어
기본 타입	boolean, byte, char, short, int, long, float, double
접근 제한자	private, protected, public
클래스와 관련된 것	class, abstract, interface, extends, implements, enum
객체와 관련된 것	new, instanceof, this, super, null
메소드와 관련된 것	void, return
제어문과 관련된 것	if, else, switch, case, default, for, do, while, break, continue
논리값	true, false
예외 처리와 관련된 것	try, catch, finally, throw, throws
기타	package, import, synchronized, final, static

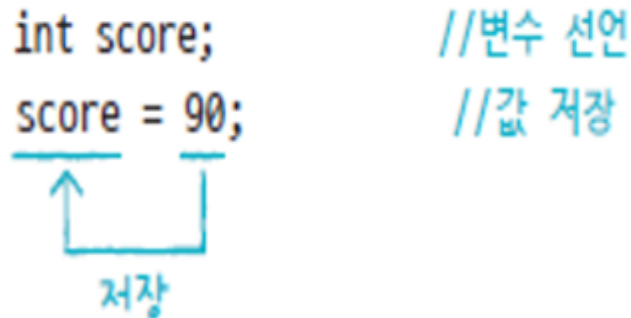


## 2. 값 저장

### ❖ 값을 저장할 경우 대입 연산자 (=) 사용

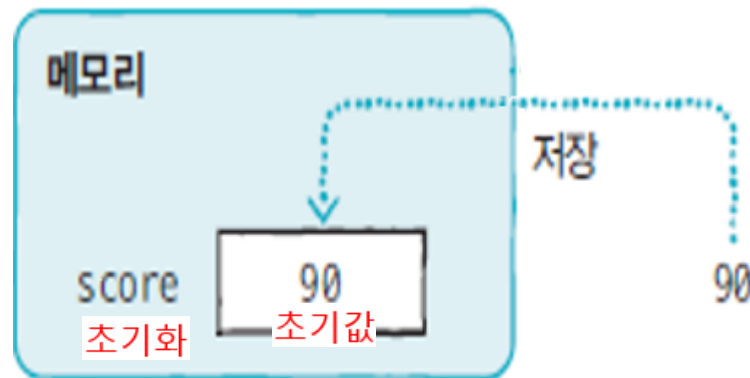
- 변수를 선언하고
- 대입 연산자를 사용해서 오른쪽의 값을 왼쪽의 변수에 저장

```
int score;           //변수 선언
score = 90;          //값 저장
```



### ❖ 변수 초기화

- 변수에 최초로 값이 저장될 때
- 메모리에 변수가 생성
- 이것을 변수 초기화라 하고
- 이 때의 값을 초기값이라고 함



## 2. 값 저장

- 초기화를 하지 않은 변수는 메모리에서 값을 읽을 수 없음

```
int value;           //변수 value 선언  
int result = value + 10; //변수 value 값을 읽고 10을 더해서 변수 result에 저장
```



```
int value = 30;      //변수 value가 30으로 초기화됨  
int result = value + 10; //변수 value 값(30)을 읽고 10을 더해서 변수 result에 저장
```

실습: VariableInitializationExample.java





### 3. 변수 사용

❖ 변수 사용: 변수의 값을 이용해서, 출력문이나 연산식을 수행하는 것

- 예시

```
int hour = 3;  
int minute = 5;
```

- 출력문에서 변수 값 사용

```
System.out.println(hour + "시간 " + minute + "분"); //변수 hour와 minute 값을 출력: 3시간 5분
```

- 연산식 내부에서 변수 값 사용

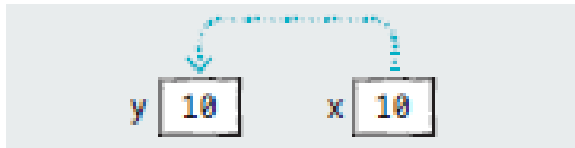
```
int totalMinute = (hour*60) + minute;
```

실습: VariableUseExample.java



### 3. 변수 사용

❖ 변수 값 복사: 변수의 값을 다른 변수에 저장



```
int x = 10;    //변수 x에 10을 저장
```

```
int y = x;     //x에 저장된 값을 변수 y에 복사(저장)
```

실습: VariableExchangeExample.java

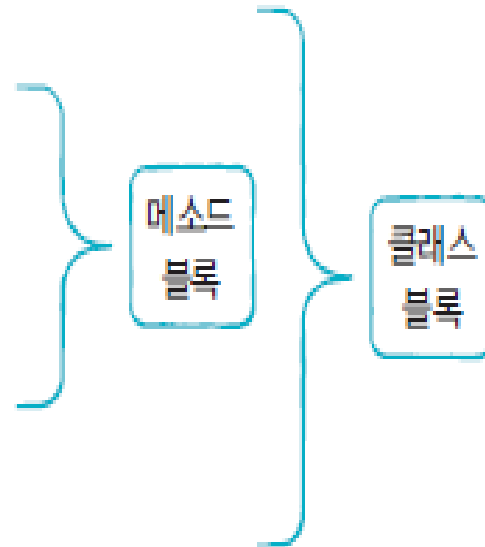


## 4. 변수 사용 범위

### ❖ 로컬 변수 (Local Variable)

- 메소드 블록 내에서 선언된 변수를 로컬 변수라고 함

```
public class VariableExample {  
    public static void main(String[] args) {  
        int value = 10;           //로컬 변수 value  
        int sum = value + 20;      //로컬 변수 sum  
        System.out.println(sum);  
    }  
}
```



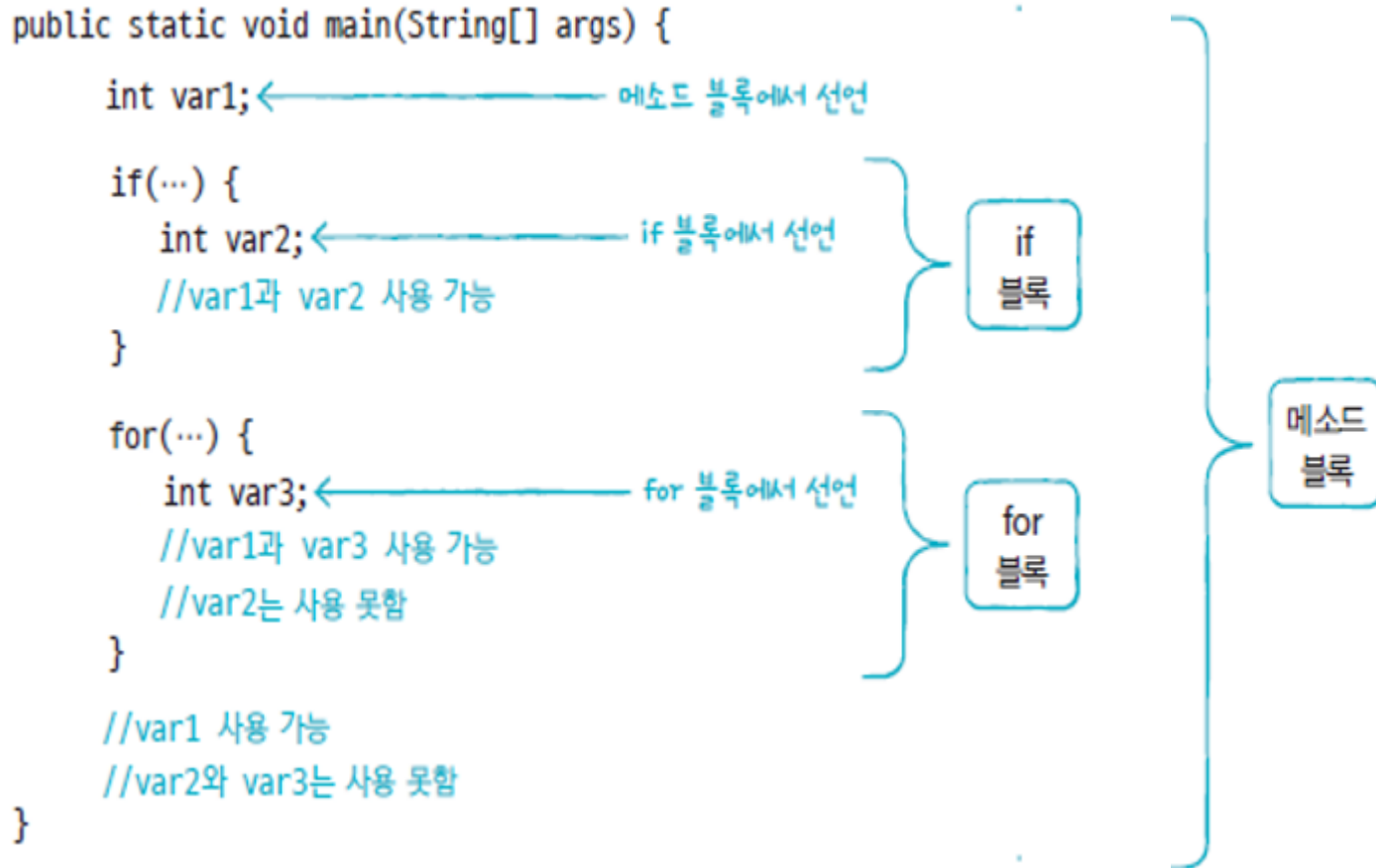
이 위치에서는 value와 sum 변수는 사용 못함

- 로컬 변수는 메소드 블록 내에서만 사용되고 메소드 실행이 끝나면 자동 삭제됨



## 4. 변수 사용 범위

- 로컬 변수는 해당 중괄호 블록 내에서만 사용 가능



## 4. 변수 사용 범위

- 다음 사항에 주의
  - 변수가 어떤 범위에서 사용될 것인지 고려하여 선언 위치 결정할 것
  - 메소드 블록 전체에서 사용하려는 경우 메소드 블록 첫머리에 선언
  - 특정 블록 내부에서만 사용하려는 경우 해당 블록 내에 선언

실습: VariableScopeExample.java



## 5. 키워드로 끝내는 핵심 포인트

- **변수**: 값을 저장할 수 있는 메모리 번지에 붙인 이름. 변수를 통해 프로그램은 메모리 번지에 값을 저장하고 읽을 수 있다.
- **변수 선언**: 변수에 어떤 타입의 데이터를 저장할지 그리고 변수 이름이 무엇인지를 결정하는 것
- **변수 사용**: 변수의 값을 읽거나 변경하는 것. 변수는 출력문이나 연산식 내부에서 사용되어 변수에 저장된 값을 출력하거나 연산에 사용한다.
- **변수 사용 범위** : 변수는 자신이 선언된 위치에서 자신이 속한 블록 내부까지만 사용이 가능하고 밖에서는 사용할 수 없다.





Thank You!