

# 변수

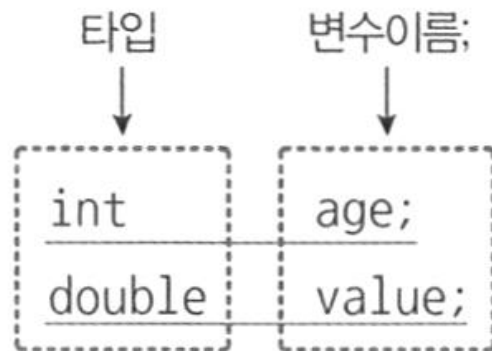
# 변수

## ❖ 변수(Variable)

- 하나의 값을 저장할 수 있는 메모리 공간
- 프로그램에 의해서 수시로 값이 변동될 수 있음
- 복수 개의 값을 저장할 수 없고, 하나의 값만 저장

## ❖ 변수의 선언

타입    변수이름



//정수(int)값을 저장할 수 있는 age 변수 선언

//실수(double)값을 저장할 수 있는 value 변수 선언

# 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

# 변수

---

## ❖ 변수의 사용

- 변수값 저장

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

초기값은 변수를 선언함과 동시에 줄 수도 있다.

```
int score = 90;
```

# 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

# 변수

---

## ❖ 예제: LiteralExample.java

```
public class LiteralExample {  
    public static void main(String[] args) {  
        int var1 = 10;  
        System.out.println(var1);  
  
        int var2 = 010;  
        System.out.println(var2);  
  
        int var3 = 0x10;  
        System.out.println(var3);  
    }  
}
```

## 변수

### ❖ 특수문자 리터럴

이스케이프 문자	용도	유니코드
<code>\t</code>	수평 탭	0x0009
<code>\n</code>	줄 바꿈	0x000a
<code>\r</code>	리턴	0x000d
<code>\"</code>	" (큰따옴표)	0x0022
<code>\'</code>	' (작은따옴표)	0x0027
<code>\\</code>	\	0x005c
<code>\u16진수</code>	16진수에 해당하는 유니코드	0x0000 ~ 0xffff

# 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
```



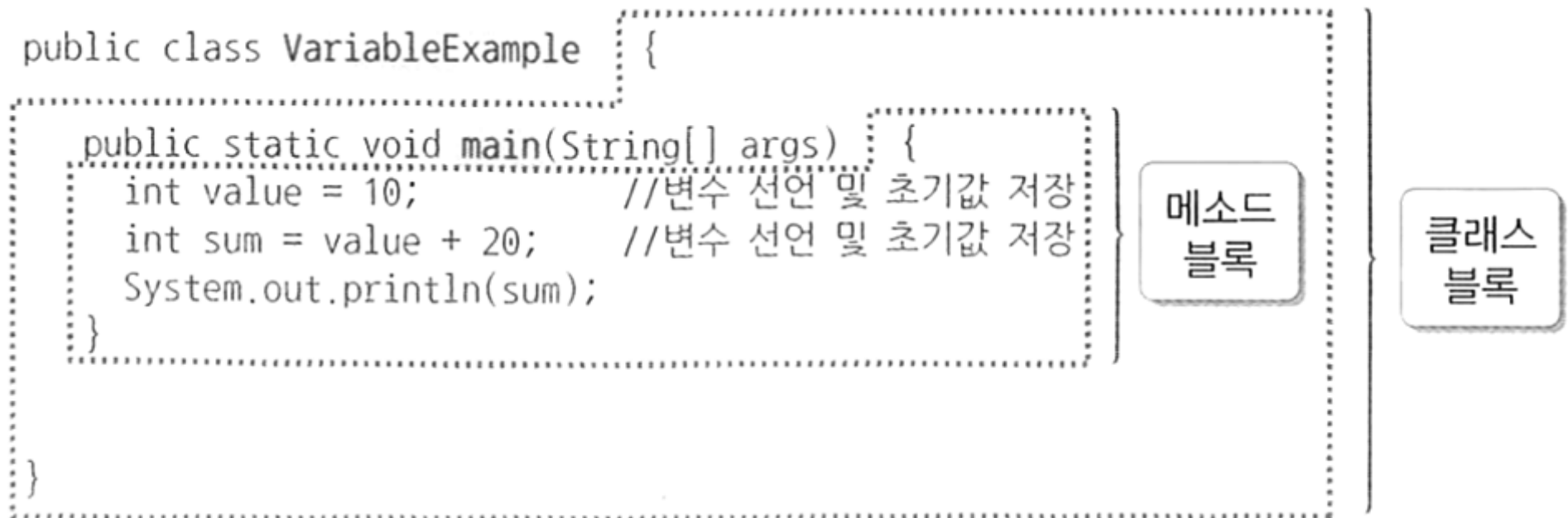
## ❖ 예제: VariableExample.java

```
public class VariableExample {  
    public static void main(String[] args) {  
        // 10을 변수 value의 초기값으로 저장  
        int value = 10;  
  
        // 변수 value 값을 읽고 10을 더하는 산술 연산을 수행  
        // 연산의 결과값을 변수 result의 초기값으로 저장  
        int result = value + 10;  
  
        // 변수 result 값을 읽고 콘솔에 출력  
        System.out.println(result);  
    }  
}
```

# 변수

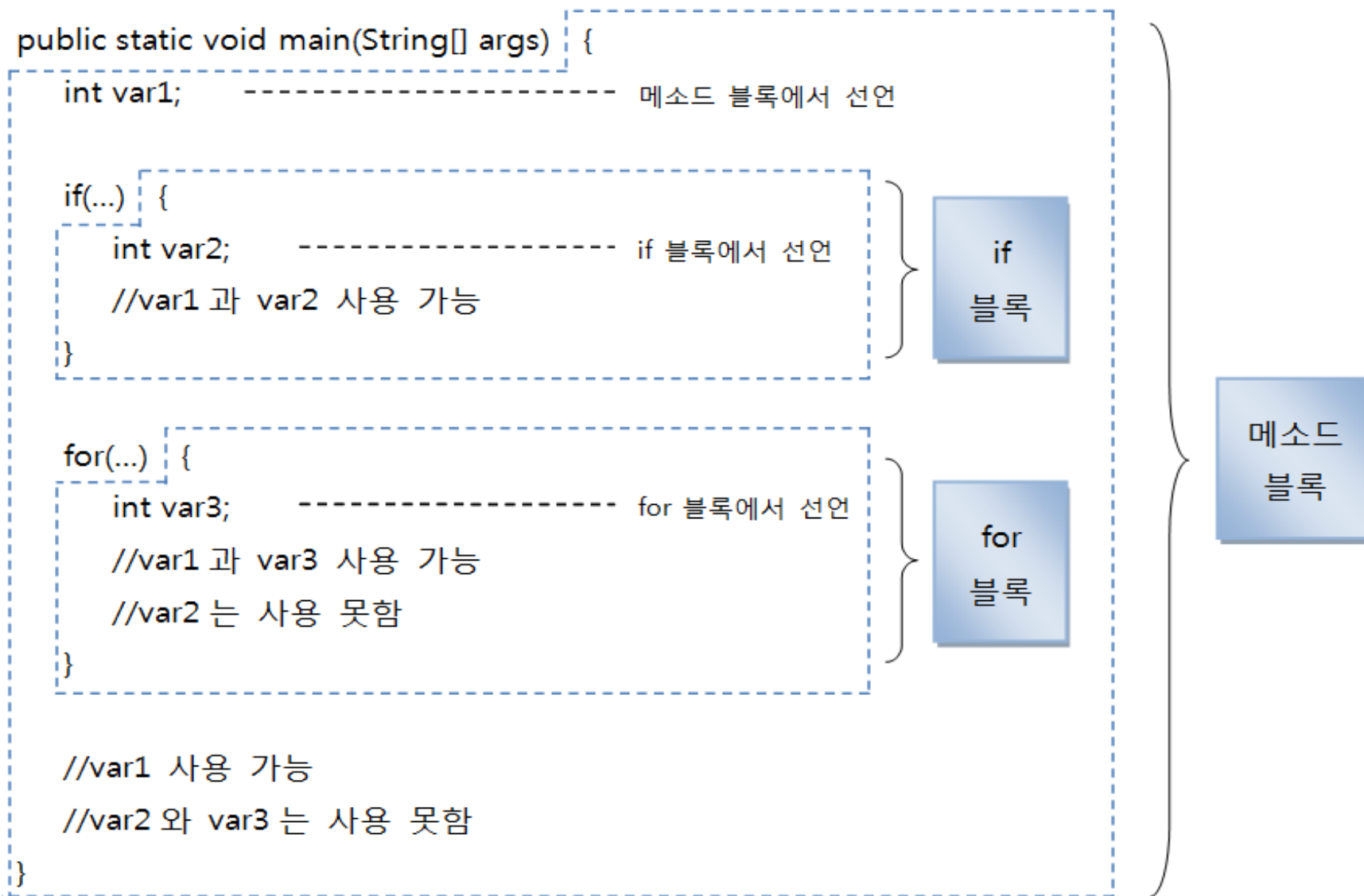
## ❖ 변수의 사용 범위

- 변수는 중괄호 블록 {} 내에서 선언되고 사용



# 변수

## ❖ 변수의 사용 범위



# 변수

---

## ❖ 예제: VariableScopeExample.java

```
public class VariableScopeExample {  
    public static void main(String[] args) {  
        int var1;  
  
        if (true) {  
            int var2;  
            var1 = 10;  
            var2 = 20;  
        }  
  
        var1 = 10;  
        // var2 = 20;  
        for (int i = 0; i < 1; i++) {  
            int var3;  
            var1 = 10;  
            // var3 = 30;  
        }  
  
        var1 = 10;  
        // var3 = 30;  
    }  
}
```

# 변수

---

## ❖ 예제: VariableScopeExample2.java

```
public class VariableScopeExample2 {  
    public static void main(String[] args) {  
        int v1 = 15;  
        if (v1 > 10) {  
            int v2;  
            v2 = v1 - 10;  
        }  
  
        // int v3 = v1 + v2 + 5;  
        //v2 변수를 사용할 수 없기 때문에 컴파일 에러가 생김  
    }  
}
```