



2장 변수와 타입

강사 김영석



CONTENT

001

데이터 타입

002

타입 변환

003

변수

001 변수

✓ 변수란 ?

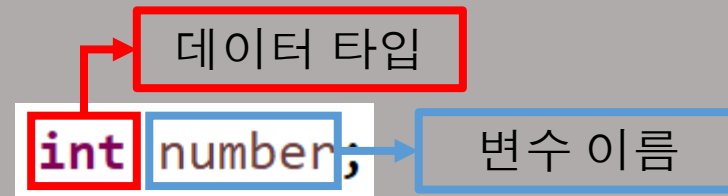
- 하나의 값을 저장할 수 있는 메모리 공간

✓ 변수 선언

- 값을 저장할 수 있는 메모리 공간의 이름과 데이터 타입을 규정하는 것

✓ 변수 선언 방법

- [데이터 타입] [변수 이름]



✓ 변수 이름 명명 규칙

작성 규칙	예
첫번째 글자는 문자이거나 '\$', '_' 여야 하고 숫자로 시작할 수 없다.	가능 : price, \$price, _price 불가능 : 1price, @price, \$#price
영어 대소문자가 구분 된다.	firstname 과 firstName 은 다른 변수
첫 문자는 영어 소문자로 시작하고, 다른 단어가 연결 될 경우 첫 문자는 대문자로 한다.	maxSpeed, firstName, carBodyColor
문자 수(길이)의 제한은 없다	
자바 예약어는 사용할 수 없다.	class, import, public, static 등

✓ 변수의 사용

- 변수값 저장

- 선언 후 값 저장

```
int number;    // 변수 선언  
number = 100;  // 값 저장
```

- 선언과 동시에 값 저장

```
int number = 100;    // 변수 선언과 값 저장
```


✓ 변수의 사용

■ 변수값 읽기

➤ 변수는 초기화가 되어야 읽기 가능

```
int number;           // 변수 선언 - 초기화 안됨  
int result = number + 10; // 초기화 되지 않아 오류 발생
```

```
int number = 30;       // 변수 선언 - 초기화  
int result = number + 10; // 초기화 되어 오류 없음
```



✓ 리터럴 (Literal)

- 변수 및 상수에 저장되는 값
- 소스 코드 내에서 직접 입력된 변수의 초기값을 의미
- 종류

```
int number = 30;           //정수 리터럴
double doubleNumber = 1.1; //실수 리터럴
char character = 'A';      //문자 리터럴
String str = "Hello";     //문자열 리터럴
Boolean bl = true;        //논리 리터럴
```

✓ 변수의 사용 범위

- 변수는 중괄호 블록 {} 내에서 선언되고 사용

```
public static void main(String[] args) {  
    int value1; // 메소드 블록에서 사용 가능  
  
    if(...) {  
        int value2; // if 블록에서 사용 가능  
        // value1 과 value2 는 사용 가능  
    }  
  
    for(...) {  
        int value3; // for 블록에서 사용가능  
        // value1 과 value3 은 사용 가능  
        // value2 는 사용 불가능  
    }  
  
    // value1 은 사용 가능  
    // value2 와 value3 은 사용 불가능  
}
```


002 데이터 타입

✓ 기본 타입

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

값의 종류	기본 타입	메모리 사용 크기	저장되는 값의 범위
정수	byte	1byte	-128 ~ 127
	char	2byte	0 ~ 65,535 (문자 표현)
	short	2byte	-32,768 ~ 32,767
	int	4byte	-2,147,483,648 ~ 2,147,483,647
	long	8byte	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
실수	float	4byte	-3.4E38 ~ 3.4E38
	double	8byte	-1.7E308 ~ 1.7E308
논리	boolean	1byte	True, false

003 타입 변환

✓ 타입 변환

- 데이터 타입을 다른 타입으로 변환하는 것

- byte ⇄ int , int ⇄ double

- 종류

- 자동(묵시적) 타입 변환 : Promotion

- 강제(명시적) 타입 변환 : Demotion

✓ 자동 타입 변환

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

자동 타입 변환



byte < short < int < long < float < double

✓ 강제 타입 변환

- 큰 타입은 작은 타입으로 형을 변환하기 위해서는 강제 변환해야 가능

작은크기 타입 = (작은크기타입)큰크기 타입

강제 타입 변환



byte < short < int < long < float < double

```
int value1 = 10;  
byte value2 = (byte)value1;
```

✓ 연산식에서 자동 타입 변환

- 연산은 같은 타입의 피연산자간에만 수행
 - 서로 다른 타입의 피연산자는 같은 타입으로 변환
 - 두 피연산자 중 크기가 큰 타입으로 자동 변환

```
int number = 10;  
double doubleNumber = 4.0;  
double result = number / doubleNumber;  
// number가 double 타입으로 자동 형 변환  
// result 에 2.5가 저장
```

```
int number = 10;  
double doubleNumber = 4.0;  
double result = number / (int)doubleNumber;  
// doubleNumber가 int 타입으로 강제 형 변환  
// result 에 2.0가 저장
```


A photograph of a server room with rows of server racks on both sides of a central aisle. The racks have glass doors and internal components are visible, with many small blue lights glowing. The ceiling has several long, rectangular light fixtures. The overall atmosphere is dimly lit, emphasizing the blue light from the servers.

수고하셨습니다.