

열혈 Java 프로그래밍

Chapter 02. 변수와 자료형

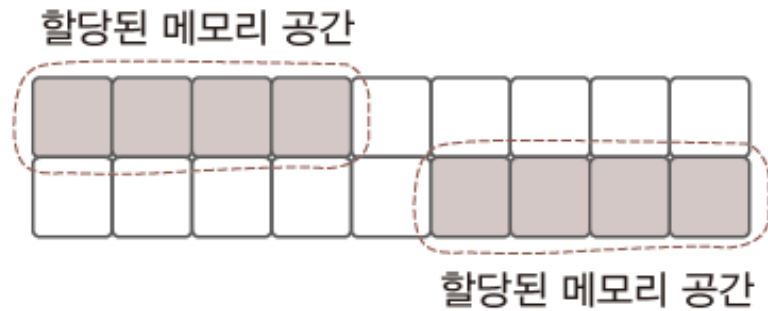
02-1.

변수의 이해와 활용

메모리 공간의 활용을 위해 필요한 변수

▶ 변수(Variable)

- 메모리 공간의 활용을 위한 도구
- 메모리 공간의 **할당**과 **접근**을 위해 필요한 도구
- 변수의 선언은 '메모리 공간의 할당'으로 이어진다.



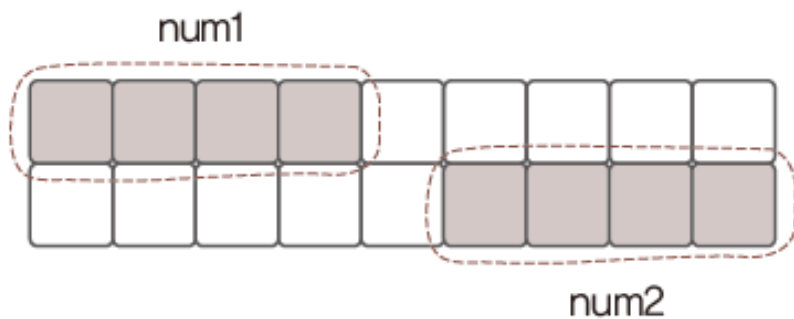
메모리 공간 할당의 예

▶ 변수의 선언을 통해 결정하는 것 두 가지!

- 변수의 이름
- 변수의 용도

int num1; ◁ 변수 num1의 선언

int num2; ◁ 변수 num2의 선언



◆ UseVariable.java

```
1. class UseVariable {  
2.     public static void main(String[] args) {  
3.         int num1;    // 변수 num1의 선언  
4.         num1 = 10;   // 변수 num1에 10을 저장  
5.  
6.         int num2 = 20; // 변수 num2 선언과 동시에 20으로 초기화  
7.         int num3 = num1 + num2; // 두 변수 값을 대상으로 덧셈  
8.         System.out.println(num1 + " + " + num2 + " = " + num3);  
9.     }  
10. }
```

자바에서 = 은 '같다'가 아닌 '대입'의 의미이다.

대입은 오른쪽에 있는 값을 왼쪽으로!



```
C:\JavaStudy>java UseVariable  
10 + 20 = 30  
C:\JavaStudy>
```

변수 활용의 예

자료형	데이터	크 기	표현 가능 범위
boolean	참과 거짓	1 바이트	true, false
char	문자	2 바이트	유니코드 문자
byte	정수	1 바이트	-128 ~ 127
short		2 바이트	-32,768 ~ 32,767
int		4 바이트	-2,147,483,648 ~ 2,147,483,647
long		8 바이트	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
float	실수	4 바이트	$\pm(1.40 \times 10^{-45} \sim 3.40 \times 10^{38})$
double		8 바이트	$\pm(4.94 \times 10^{-324} \sim 1.79 \times 10^{308})$

자바에서 기본적으로 제공하는 자료형이라 하여 '기본 자료형(Primitive Data Type)'이라 한다.

자료형의 종류와 구분

◆ VariableDecl.java

```
1. class VariableDecl {  
2.     public static void main(String[] args) {  
3.         double num1, num2;    // 두 개의 변수 동시 선언  
4.         double result;  
5.         num1 = 1.0000001;  
6.         num2 = 2.0000001;  
7.         result = num1 + num2;  
8.         System.out.println(result);  
9.     }  
10. }
```

명령 프롬프트

```
C:\JavaStudy>java VariableDecl  
3.0000001999999997
```

```
C:\JavaStudy>_
```

기대하는 값 3.00000002 가 출력되지 않았다.

이유는 실수 표현에 오차가 존재하기 때문이다.

다양한 자료형 활용의 예

01

자바는 대소문자를
구분한다.

02

변수의 이름은
숫자로 시작할 수
없다.

03

\$과 _ 이외의
특수문자는 변수의
이름에 사용할 수
없다.

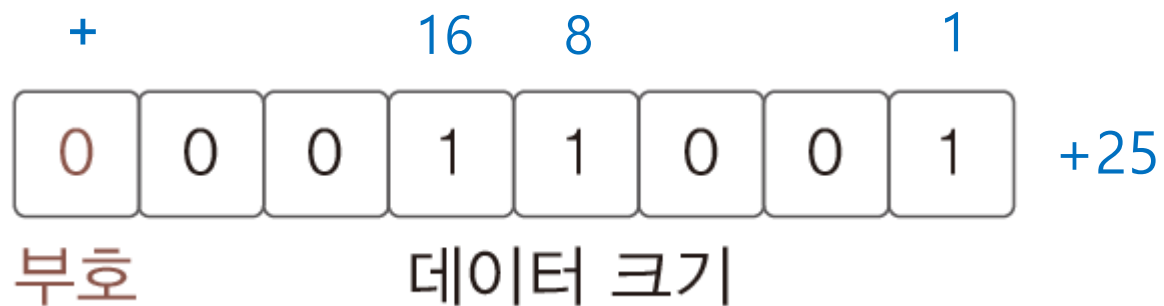
04

키워드는 변수의
이름으로 사용할 수
없다.

변수의 이름을 짓는데 있어서의 제약사항

02-2.

정수의 표현 방식 이해하기



- 부호 비트가 0이면 양의 정수, 1이면 음의 정수
- 부호 비트가 0이면, 나머지 비트들은 값의 크기를 결정

컴퓨터가 양의 정수를 표현하는 방식



-25가 맞을까?

0	0	0	1	1	0	0	1	
+	1	0	0	1	1	0	0	1

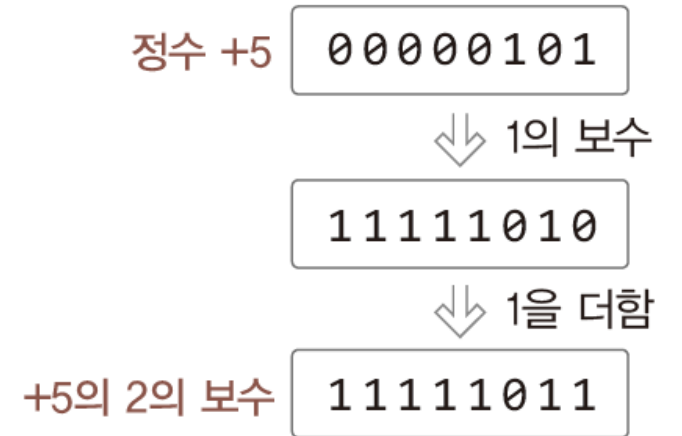
~ 0이 되어야 하는데 ~

양의 정수 표현법 기반으로 음의 정수를 표현한다면?

음의 정수를 표현하는 방법

- 양의 정수의 이진수 표현에 2의 보수를 취한 결과를 음의 정수로 표현한다.

- 이 경우, 임의의 양의 정수가 있을 때,
이와 절댓값이 같은 음의 정수의 합은 0이 된다.



$$\begin{array}{r} 00000101 \\ + 11111011 \\ \hline 1\ 00000000 \end{array}$$

올림 수 버림

02-3.

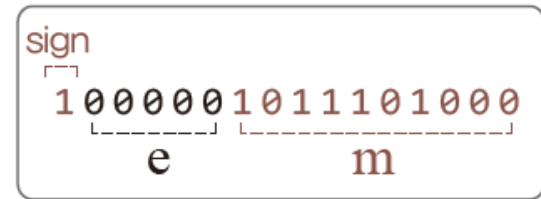
실수의 표현 방식 이해하기

실수의 표현 방식 이해

- 정수와 달리 실수는 오차 없이 표현이 불가능
- 따라서 정밀도를 낮추고 표현할 수 있는 값의 범위 넓힘
- 실수 표현 방법의 기준 『IEEE 754』

$$\pm (1.m) \times 2^{e-127}$$

↑ 이 수식에 반영



```
double num1 = 1.00000001;
```

```
double num2 = 2.00000001;
```

num1과 num2에는

최대한 가까운 실수의 표현이 저장된다.

02-4.

자바의 기본 자료형

byte

1 byte

short

2 byte

int

4 byte

long

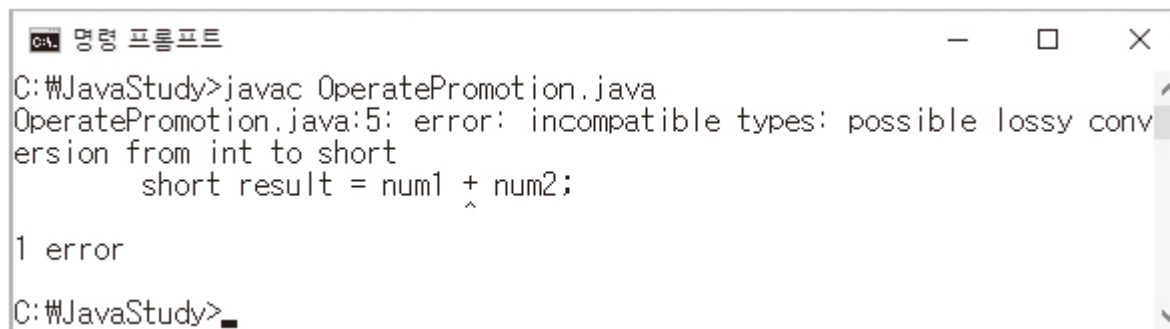
8 byte

변수의 자료형 결정은 '해당 변수에 값을 저장 및
참조하는 방식의 결정'을 의미한다.

정수 자료형

◆ OperatePromotion.java

```
1. class OperatePromotion {  
2.     public static void main(String[] args) {  
3.         short num1 = 11;  
4.         short num2 = 22;  
5.         short result = num1 + num2;  
6.         System.out.println(result);  
7.     }  
8. }
```



A screenshot of a Windows command prompt window titled "명령 프롬프트". The window shows the command `C:\JavaStudy>javac OperatePromotion.java` and the resulting error message: `OperatePromotion.java:5: error: incompatible types: possible lossy conversion from int to short`. The error points to the line `short result = num1 + num2;` where the `+` operator is highlighted. Below the error message, it says "1 error". The prompt ends with `C:\JavaStudy>_`.

컴파일 에러는 정수형 덧셈 시 자료형에 상관없이 int형 덧셈을 진행함을 의미한다.

short형 변수와 int형 변수 중 하나를 선택한다면?

float

4 byte

double

8 byte

float	실수	4 바이트	$\pm(1.40 \times 10^{-45} \sim 3.40 \times 10^{38})$
double		8 바이트	$\pm(4.94 \times 10^{-324} \sim 1.79 \times 10^{308})$

float와 double 사이의 자료형 선택 기준은 **정밀도!**

실수 자료형

	D5D	D5E	D5F	D60	D61	D62	D63	D64	D65
0	헐 D5D0	헝 D5E0	험 D5F0	혀 D600	헝 D610	헞 D620	헟 D630	홀 D640	혹 D650
1	헝 D5D1	헞 D5E1	헟 D5F1	헙 D601	헚 D611	헛 D621	헜 D631	홉 D641	홉 D651
2	헞 D5D2	헟 D5E2	헟 D5F2	헛 D602	헜 D612	헝 D622	헞 D632	홉 D642	홉 D652
3	헟 D5D3	헟 D5E3	헟 D5F3	헜 D603	헝 D613	헞 D623	헟 D633	홉 D643	홉 D653
4	헟 D5D4	헞 D5E4	헞 D5F4	헞 D604	헟 D614	헞 D624	헟 D634	홉 D644	화 D654
5	헟 D5D5	헟 D5E5	헞 D5F5	헟 D605	헞 D615	헟 D625	헞 D635	홉 D645	화 D655

한글 유니코드의 일부

문자 자료형

- 자바의 문자 자료형 char
- 자바는 문자를 2바이트 유니코드로 표현한다.
- 작은 따옴표로 묶어서 하나의 문자를 표시한다.
- 문자의 저장은 유니코드 값의 저장으로 이어진다.

```
char ch1 = '헐';
```

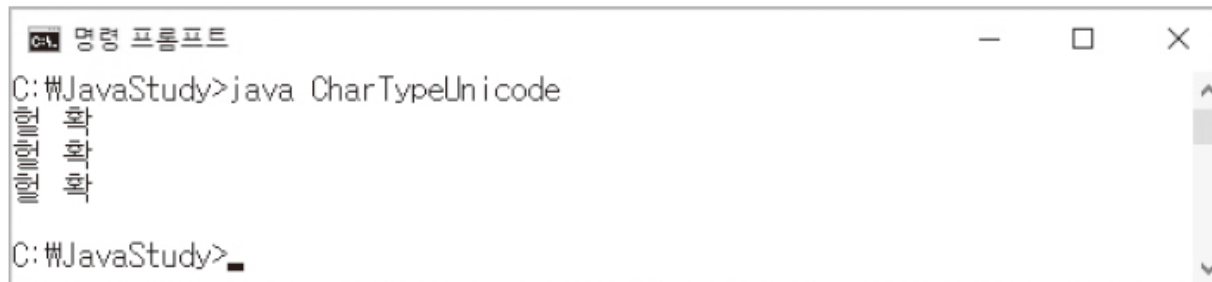
```
char ch2 = '혹';
```

문자 '헐'의 유니코드 값 D5D 와 0 의 조합 결과 D5D0

문자 '혹'의 유니코드 값 D65 와 5 의 조합 결과 D655

◆ CharTypeUnicode.java

```
1. class CharTypeUnicode {
2.     public static void main(String[] args) {
3.         char ch1 = '혈';
4.         char ch2 = '확';
5.         char ch3 = 54736;    // 문자 '혈'의 유니코드 값
6.         char ch4 = 54869;    // 문자 '확'의 유니코드 값
7.         char ch5 = 0xD5D0;
8.         char ch6 = 0xD655;
9.         System.out.println(ch1 + " " + ch2);
10.        System.out.println(ch3 + " " + ch4);
11.        System.out.println(ch5 + " " + ch6);
12.    }
13. }
```



```
C:\> 명령 프롬프트
C:\> C:\JavaStudy>java CharTypeUnicode
혈 확
혈 확
혈 확
혈 확
C:\JavaStudy>
```

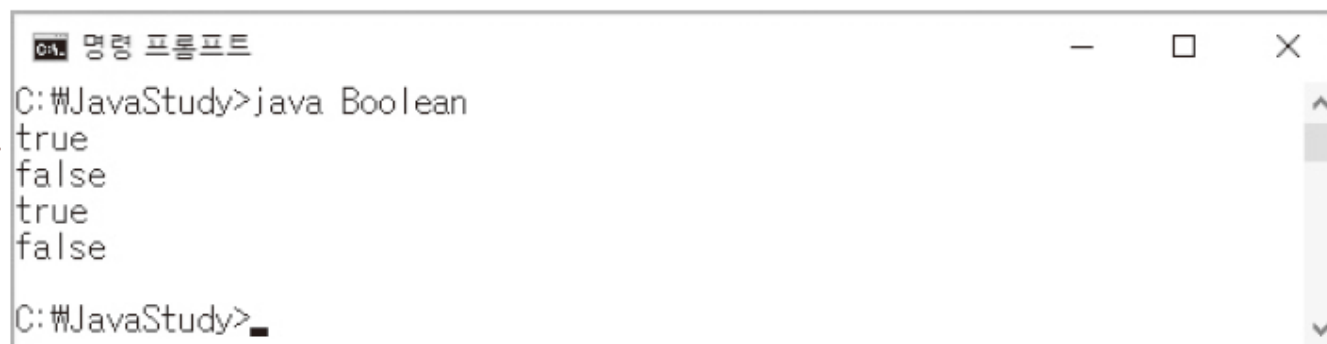
코드상에서 문자 표현의 예

◆ Boolean.java

```
1. class Boolean {
2.     public static void main(String[] args) {
3.         boolean b1 = true;
4.         boolean b2 = false;
5.         System.out.println(b1);    // b1이 지닌 값 출력
6.         System.out.println(b2);
7.
8.         int num1 = 10;
9.         int num2 = 20;
10.        System.out.println(num1 < num2);
11.        System.out.println(num1 > num2);
12.    }
13. }
```

true '참'을 의미하는 값

false '거짓'을 의미하는 값



명령 프롬프트

```
C:\JavaStudy>java Boolean
true
false
true
false

C:\JavaStudy>_
```

논리 자료형

The End

Chapter 02의 강의를 마칩니다.