

프로그래밍 기초

Chapter 02. 변수와 자료형

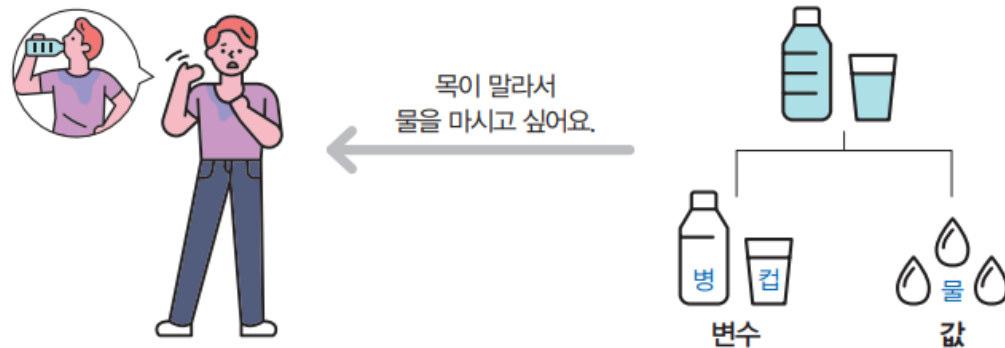
Section 01

변수

변수

■ 실생활 속 변수의 개념

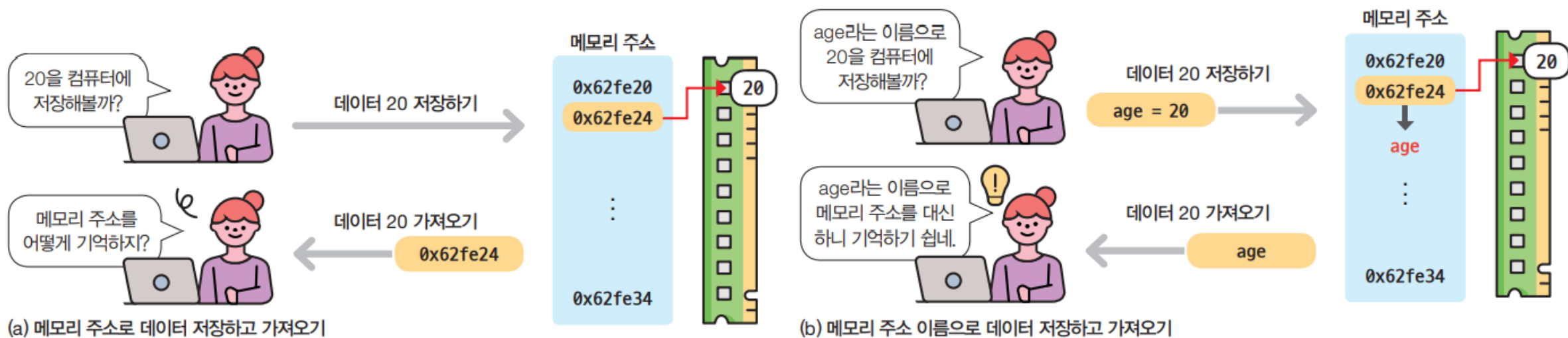
- 무더운 날 목이 너무 말라서 물을 마시려고 할 때 물은 항상 무언가에 담겨 있음
 - 변수(variable) : 물을 담고 있는 컵 또는 병
 - 값(value) : 컵 또는 병에 담겨 있는 물
- 변수의 기본 개념
 - 컵 또는 병에 담기는 것을 주스, 콜라 등으로 변경할 수도 있음
 - 이름이 지정된 변수 안의 값이 다양하게 변경될 수 있음을 의미함



변수

■ 프로그래밍 속 변수의 개념

- 변수 : 컴퓨터 프로그램이 처리할 데이터를 저장하는 저장 위치([예]컵, 병)
 - 데이터가 저장되는 메모리 주소가 복잡하기 때문에 메모리 주소 대신 메모리 주소의 이름을 지정함
 - 이러한 메모리 주소의 이름, 즉 메모리 위치의 이름을 변수라고 함
 - ✓ 여기에 저장된 값은 변경될 수 있음
- 즉, 변수는 특정 자료형의 값을 저장하는 메모리 공간의 위치를 의미함
 - 변수는 이름과 자료형을 가지고 있으며, 자료형에 따른 값이 저장됨



변수

■ 변수 선언의 기본 형식

- 변수를 사용하려면 먼저 변수를 선언해야 함

```
자료형  변수명;  
int      myVar;
```

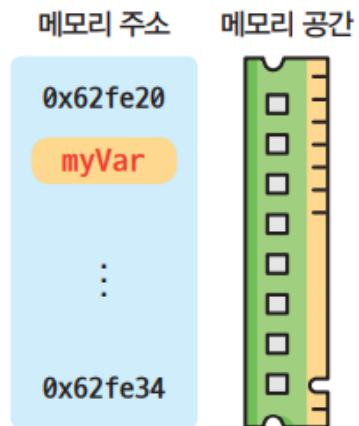
→ 자료형(data type): 변수에 저장할 수 있는 데이터의 유형

→ 변수명(variable name): 메모리 주소를 대신하는 메모리 주소의 이름을 지정

- [예] 정수형 변수 time과 speed를 선언하는 예

```
int time;  
int time, speed; // 변수의 자료형이 동일한 경우
```

int myVar
자료형 변수명

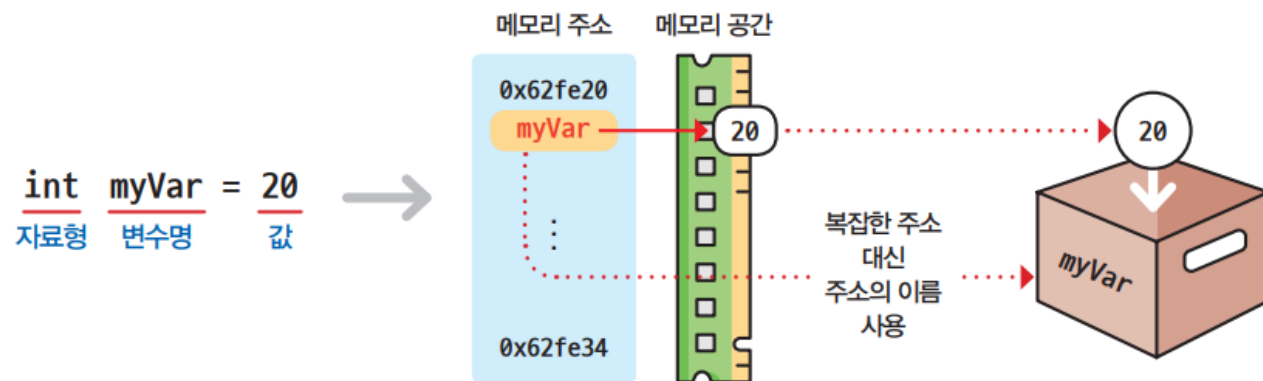


변수

■ 변수 초기화

- 지정된 메모리 위치에 값을 입력하는 것
- 변수 초기화는 변수 선언과 동시에 값을 할당하는 것

자료형 변수명 = 값;
int myVar = 20;

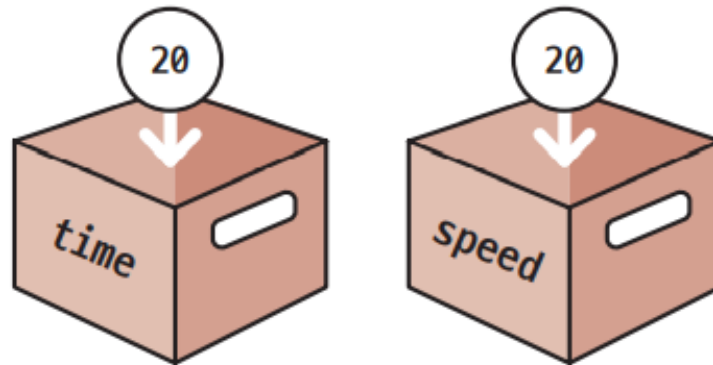


변수

■ 변수 초기화

- [예] 정수형 변수 time과 speed를 선언하는 동시에 20으로 초기화하는 예

```
int time = 20;  
int time = 20, speed = 20; // 변수의 자료형이 동일한 경우
```



변수

■ 변수명 생성 규칙

- 변수명은 영문자(대·소문자), 숫자, 밑줄(_), 달러 기호(\$)로만 구성할 수 있음
- 변수명은 숫자로 시작할 수 없음
- 변수명에는 공백을 포함할 수 없음
- 자바에서 미리 정의된 키워드를 변수명으로 사용할 수 없음

변수

■ 예제 2-1. 변수 선언하고 초기화하기

```
01 public class Variable01 {  
02     public static void main(String[] args) {  
03  
04         int days = 10;  
05         System.out .println(days); // days 변수 출력  
06  
07         int speed;  
08         speed = 20;  
09         System.out .println(speed); // speed 변수 출력  
10     }  
11 }
```

실행 결과

10

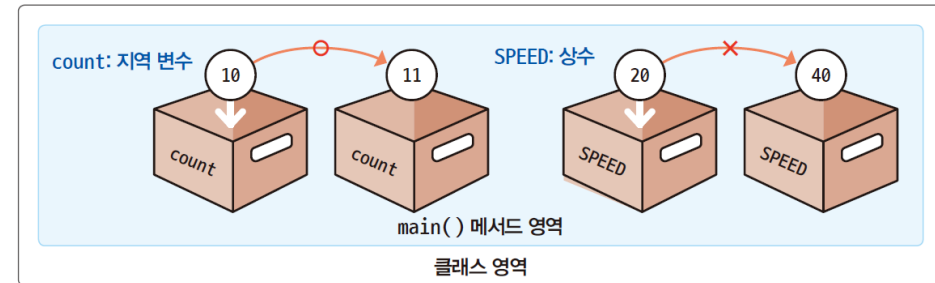
20

변수

■ 상수

- 프로그램이 실행되는 동안 메모리에 저장된 데이터를 변경할 수 없고 반드시 선언과 동시에 초기화해야 함
- 상수는 final 키워드를 이용하여 선언함

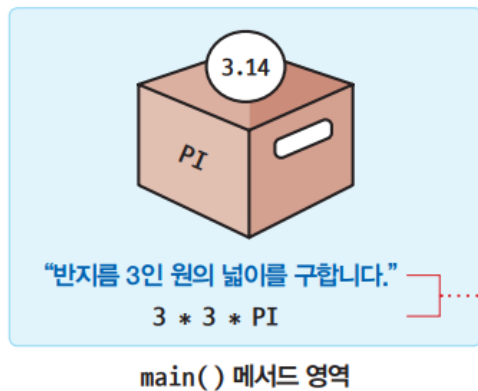
```
public class Example01 {  
    public static void main(String[] args) {  
        final int SPEED = 20;  
        int count = 10;  
        SPEED = 40;  
        count = 11;  
  
        System.out.print("상수 SPEED 값은 ");  
        System.out.println(SPEED);  
        System.out.print("변수 count 값은 ");  
        System.out.println(count);  
    }  
}
```



변수

■ 예제 2-2. 상수를 사용하여 원의 넓이 구하기

```
01 public class Variable02 {  
02     public static void main(String[] args) {  
03         final double PI = 3.14;  
04         System.out.println("반지름 3인 원의 넓이를 구합니다.");  
05         System.out.print(3 * 3 * PI);  
06     }  
07 }
```



System.out



실행 결과

반지름 3인 원의 넓이를 구합니다.
28.26

Section 02

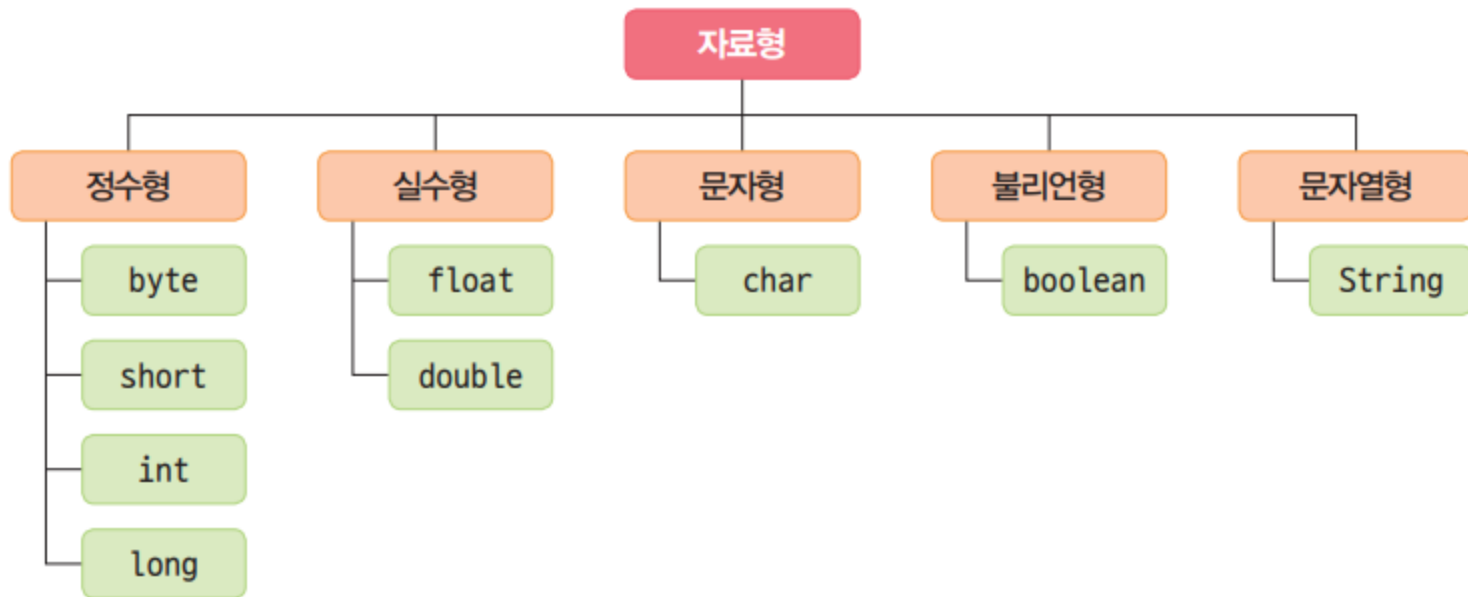
자료형

자료형

■ 자료형의 개요

• 자료형

- 프로그래밍을 할 때 사용하는 숫자, 문자열 등의 다양한 데이터 형태를 식별하는 분류로서 프로그램의 가장 기본이자 핵심적인 단위
- 자료형은 변수를 어떻게 사용할 것인지를 컴파일러나 인터프리터에 알려주기 위한 변수의 속성
- 데이터에 대해 수행할 수 있는 작업과 저장할 수 있는 값 유형을 정의



자료형

■ 정수 자료형

- 정수형은 소수 부분이 없고 음숫값을 허용하는 숫자
- byte, short, int, long 등 네 가지 유형이 있음

유형	기본값	저장 공간	범위
byte	0	1바이트	-128~127
short	0	2바이트	-32,768~32,767
int	0	4바이트	-2,147,483,648~2,147,483,647
long	0L	8바이트	-9,223,372,036,854,775,808~9,223,372,036,854,775,807

- [예]

```
byte b = 100;
```

```
short s = 123;
```

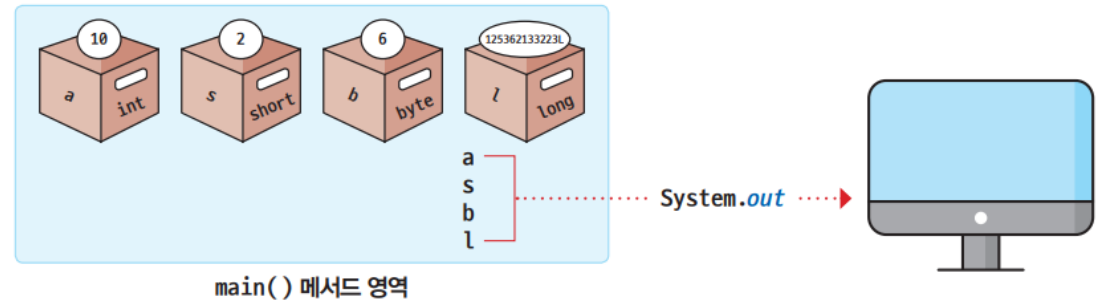
```
int in = 123543;  
int in2 = -9876345;
```

```
long l = 1234567891L;
```

자료형

■ 예제 2-3. 정수 자료형 변수 선언하고 변수값 출력하기

```
01 public class Datatype01 {  
02     public static void main(String[] args) {  
03         int a = 10;  
04         short s = 2;  
05         byte b = 6;  
06         long l = 125362133223L;  
07         System.out.println(a);  
08         System.out.println(s);  
09         System.out.println(b);  
10         System.out.println(l);  
11     }  
12 }
```



실행 결과

```
10  
2  
6  
125362133223
```

자료형

■ 실수 자료형

- 실수형은 소수 부분이 있고 음숫값을 허용하는 숫자
- float, double 등 두 가지 유형이 있음

유형	기본값	저장 공간	대략적인 범위
float	0.0f	4바이트	$1.4e-045 \sim 3.4e+038$
double	0.0d	8바이트	$4.9e-324 \sim 1.8e+308$

- [예]

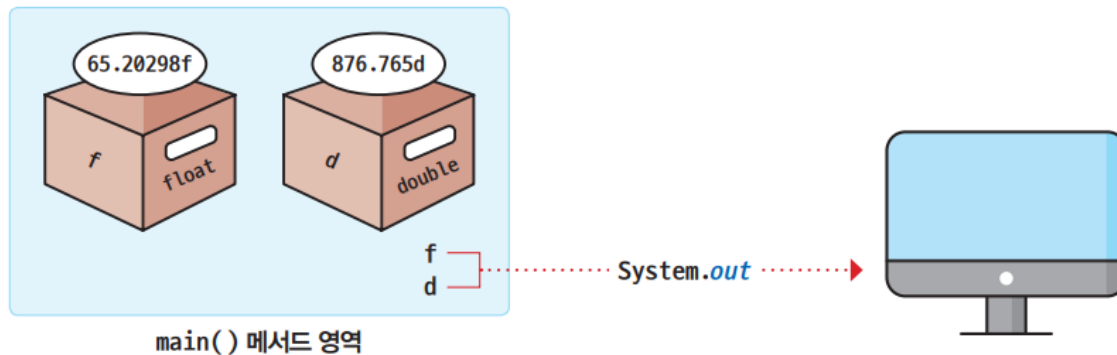
```
float pi = 12.25f;
```

```
double d = 12345.234d;
```


자료형

■ 예제 2-4. 실수 자료형 변수 선언하고 변수값 출력하기

```
01 public class Datatype02 {  
02     public static void main(String[] args) {  
03         float f = 65.20298f;  
04         double d = 876.765d;  
05         System.out.println(f);  
06         System.out.println(d);  
07     }  
08 }
```



실행 결과

```
65.20298  
876.765
```

자료형

■ 문자 자료형

- char 자료형
- 문자와 숫자 등 문자 집합의 기호를 나타내는 데 사용함
- 작은따옴표로 값을 묶어 표현함
 - [예] 'A'는 값이 65인 문자 상수
 - ✓ 단일 문자를 포함하는 문자열인 "A"와 다름

유형	기본값	저장 공간	범위
char	'0'	2바이트	\u0000(0)~\uffff(65,535)

- [예]

```
char ch1 = 88;    // X를 출력하는 코드
char ch2 = 'Y';
```

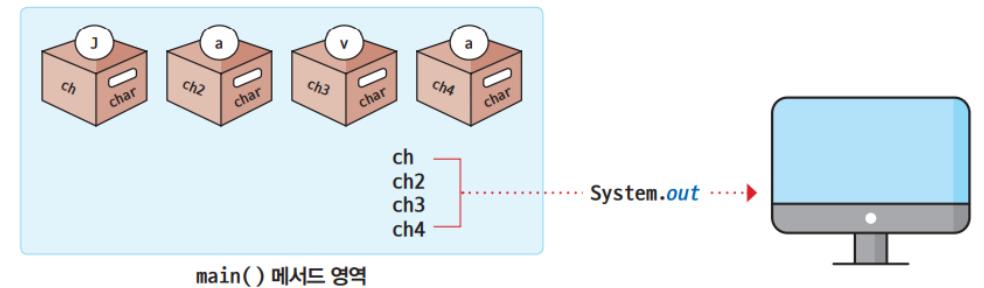
자료형

■ 예제 2-5. 실수 자료형 변수 선언하고 변수값 출력하기

```
01 public class Datatype03 {  
02     public static void main(String[] args) {  
03         char ch = 'J';  
04         char ch2 = 'a';  
05         char ch3 = 'v';  
06         char ch4 = 'a';  
07         System.out.println(ch);  
08         System.out.println(ch2);  
09         System.out.println(ch3);  
10         System.out.println(ch4);  
11     }  
12 }
```

실행 결과

J
a
v
a



main() 메서드 영역

자료형

■ 문자열 자료형

- String 자료형
- 문자열을 저장하는 데 사용함
- 큰따옴표로 값을 묶어 표현함
 - [예] “Hello” 는 문자 ‘H’, ‘e’, ‘l’, ‘l’, ‘o’로 이루어진 문자열
 - [예] “Hello World”, “Hello123”, “123”, “Name: John”
- String 자료형은 클래스에 해당하므로 반드시 대문자로 시작해야 함

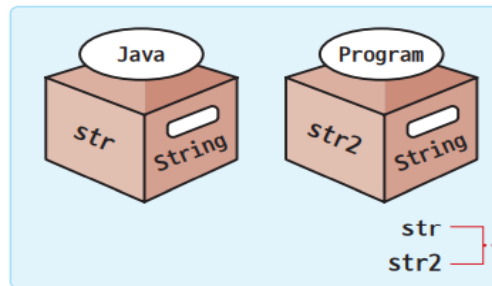
자료형

■ 예제 2-6. 문자열 자료형 변수 선언하고 변수값 출력하기

```
01 public class Datatype04 {  
02     public static void main(String[] args) {  
03         String str = "Java";  
04         String str2 = "Program";  
05  
06         System.out.println(str);  
07         System.out.println(str2);  
08     }  
09 }
```

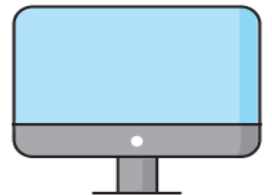
실행 결과

Java
Program



main() 메서드 영역

System.out



자료형

■ 불리언 자료형

- boolean 자료형
- 값은 true와 false, 두 가지뿐임
- if 또는 while과 같은 제어문을 제어하는 조건식에 사용함
- [예]

```
boolean flag = true;  
boolean val = false;
```

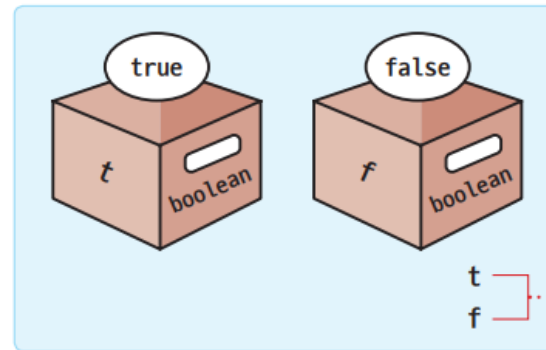
자료형

■ 예제 2-7. 불리언 자료형 변수 선언하고 변수값 출력하기

```
01 public class Datatype05 {  
02     public static void main(String[] args) {  
03         boolean t = true;  
04         boolean f = false;  
05  
06         System.out.println(t);  
07         System.out.println(f);  
08     }  
09 }
```

실행 결과

true
false



main() 메서드 영역

Section 03

출력 처리

출력 처리

■ 데이터 출력

- print() 메서드

→ 데이터를 출력하는 가장 기본적인 방법은 콘솔로 출력하는 메서드

→ print() 메서드를 사용하려면 System 클래스를 사용해야 함

메서드	설명
System.out.print()	따옴표 안의 문자열, 변수 값을 출력한다.
System.out.println()	따옴표 안의 문자열, 변수 값을 출력한 후 커서를 다음 행의 시작 부분으로 이동한다.
System.out.printf()	문자열 형식화를 제공한다(C/C++ 프로그래밍의 printf와 유사).

```
public class Example02 {  
    public static void main(String[] args) {  
        System.out.println("Java " + "Program");  
    }  
}
```

실행 결과

Java Program

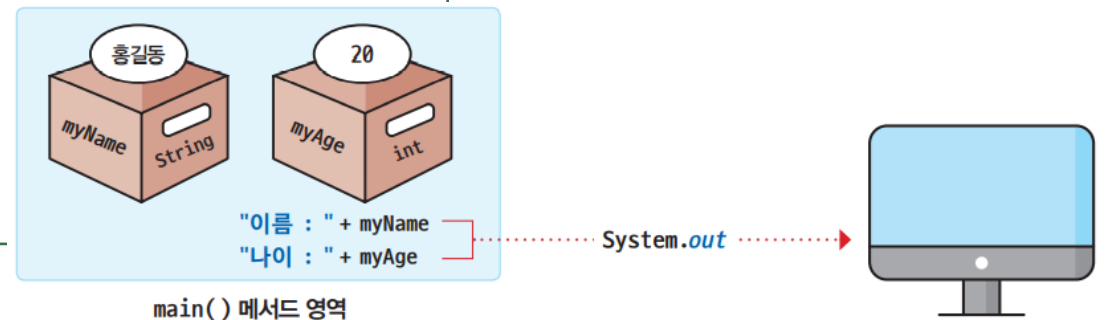
출력 처리

■ 예제 2-8. 문자열과 변수 결합하여 출력하기

```
01 public class InputOutput01 {  
02     public static void main(String[] args) {  
03  
04         String myName = "홍길동";  
05         int myAge = 20;  
06  
07         System.out.println("이름 : " + myName);  
08         System.out.println("나이 : " + myAge);  
09     }  
10 }
```

실행 결과

이름 : 홍길동
나이 : 20



출력 처리

■ 이스케이프 문자 출력

- 자바에는 이스케이프 문자를 표현하는 다양한 이스케이프 시퀀스가 있음
- [예] “HelloWn”

이스케이프 문자	설명	유니코드 값	이스케이프 문자	설명	유니코드 값
\b	백스페이스	\u0008	\"	"	\u0022
\t	탭	\u0009	\'	'	\u0027
\n	줄 바꿈	\u000a	\\	\	\u005c
\r	캐리지 리턴	\u000d			

```
public class Example03 {  
    public static void main(String[] args) {  
        System.out.println("Java\nProgram");  
    }  
}
```

실행 결과

Java
Program

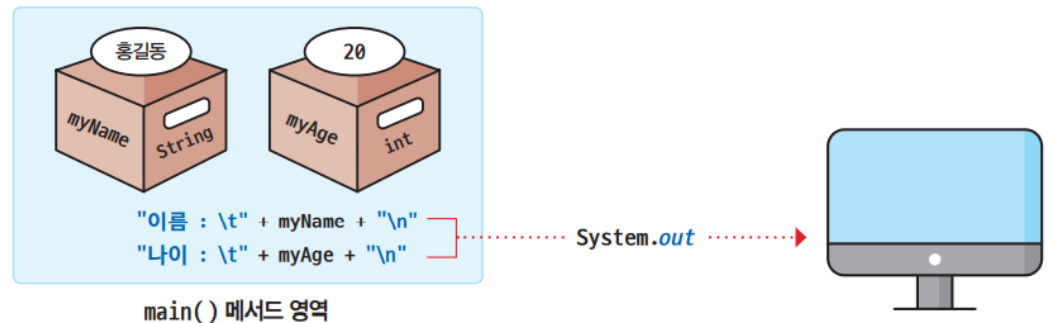
출력 처리

■ 예제 2-9. 이스케이프 문자 출력하기

```
01 public class InputOutput02 {  
02     public static void main(String[] args) {  
03  
04         String myName = "홍길동"; // 문자열 자료형  
05         int myAge = 20; // 정수 자료형  
06  
07         System.out.print("이름 : \t " + myName + " \n " );  
08         System.out.print("나이 : \t " + myAge + " \n " );  
09     }  
10 }
```

실행 결과

이름 : 홍길동
나이 : 20



프로그래밍기초

Copyright © Lee Seungwon Professor
All rights reserved.

<Q&A : lsw@kopo.ac.kr>