

## 9주 3강

# 기본 자바 소개 2



# 이번 주차에는...

## 기본 자바 소개 2

- 변수
- 데이터 형식

# 1. System.out의 기본(1)

- 화면에 내용을 출력

- `System.out.println()` : 괄호 안의 내용을 출력한 후 한 행을 띄움
- `System.out.print()` : 괄호 안의 내용을 출력한 후 한 행을 띄지 않고 유지
- `System.out.printf()` : 서식을 지정해서 출력

## 2. System.out의 기본(2)

- System.out.printf( ) 메소드의 기본적인 사용법

```
System.out.println("안녕하세요?");  
System.out.println("Java입니다.");
```

실행 결과 ▶

안녕하세요?  
Java입니다.

```
System.out.print("안녕하세요?");  
System.out.print("Java입니다.");
```

실행 결과 ▶

안녕하세요?Java입니다.

### 3. System.out의 기본(3)

```
System.out.printf("안녕하세요?");
```

실행 결과 ►

안녕하세요?

- TIP : println( )에서 ln은 line feed의 약자로 행을 넘긴다는 의미
- System.out.printf( )에서 f는 format의 약자로 서식을 지정한다는 의미

```
System.out.printf("100");
```

실행 결과 ►

100

```
System.out.printf("%d", 100);
```

실행 결과 ►

100

- 위 첫 번째 System.out.printf("100")의 결과 100은 숫자 100이 아닌 글자 100(일영영)임
- 두 번째 System.out.printf("%d", 100)의 결과 100은 숫자 100을 의미. 서식(%d)이 지정된 '숫자'는 그대로 숫자의 의미임

## 4. System.out의 기본(4)

### 실습 3-1 System.out.printf() 메소드 사용 예 1

```
01 public class Ex03_01 {  
02     public static void main(String[] args) {  
03         System.out.printf("100+100"); ----- 모두 글자로 취급한다.  
04         System.out.printf("\n"); ----- System.out.printf()는 행이 넘어가지 않으므로 강제로  
05         System.out.printf("%d", 100 + 100); ----- 숫자와 계산해서 결과를 출력한다.  
06         System.out.printf("\n"); ----- System.out.printf()는 행이 넘어가지 않으므로 강제로  
07     }  
08 }
```

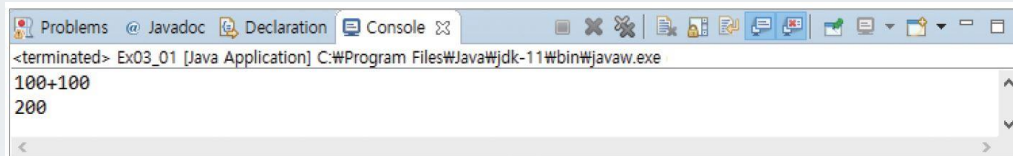


그림 3-1 실행 결과

- 5행의 “%d”는 정수(decimal)를 의미, 서식의 개수와 큰따옴표 뒤에 나오는 숫자(또는 문자)의 개수가 같아야 함

## 5. System.out의 기본(5)

### 실습 3-2 System.out.printf() 메소드 사용 예 2

```
01 public class Ex03_02 {  
02     public static void main(String[] args) {  
03         System.out.printf("%d", 100, 200); ----- %d는 1개, 숫자는 2개이다.  
04         System.out.printf("\n");  
05         System.out.printf("%d %d", 100); ----- %d는 2개, 숫자는 1개이다.  
06         System.out.printf("\n");  
07     }  
08 }
```

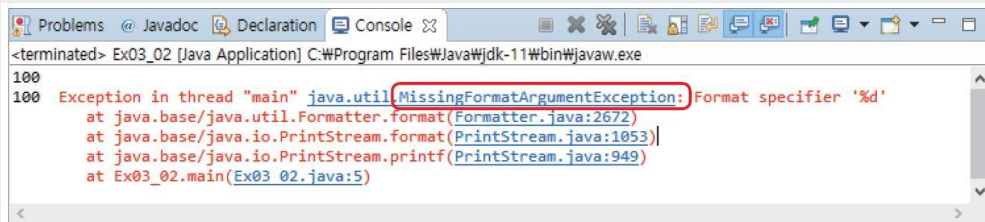


그림 3-2 실행 결과

System.out.printf("%d %d", 100, 200);

그림 3-3 서식과 숫자의 대응

## 6. System.out의 기본(6)

### 정수 외에 자주 사용되는 서식

#### 실습 3-3 서식을 사용한 출력의 예 1

```
01 public class Ex03_03 {  
02     public static void main(String[] args) {  
03         System.out.printf("%d / %d = %d", 100, 200, 0.5); ----- %d가 3개, 숫자도 3개이다.  
04         System.out.printf("\n");  
05     }  
06 }
```

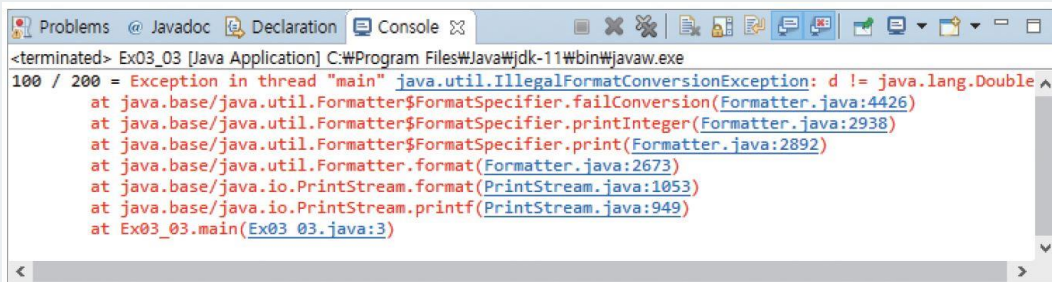


그림 3-4 실행 결과



## 7. System.out의 기본(7)

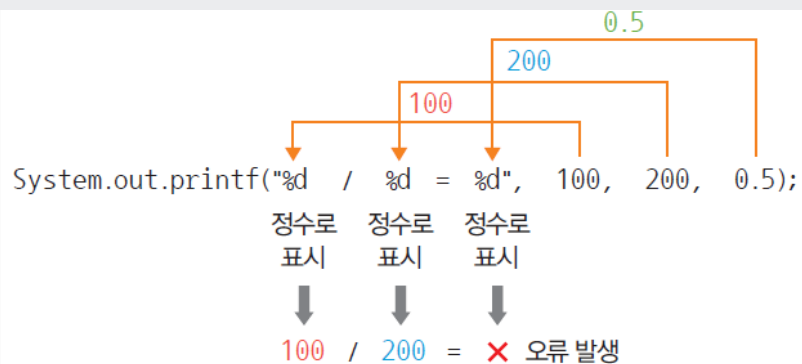


그림 3-5 서식과 숫자의 불일치 상황

표 3-1 System.out.printf()의 대표적 서식

서식	설명	값의 예
%d, %x, %o	정수(10진수, 16진수, 8진수)	10, 100, 1234
%f	실수(소수점이 있는 수)	0.5, 1.0, 3.14
%c	문자. 반드시 한 글자이고 작은따옴표(' ')로 묶여 있어야 함	'a', 'b', 'F'
%s	문자열. 한 글자 이상이고 큰따옴표(" ")로 묶여 있어야 함	"안녕", "abcdefg", "a"

## 8. System.out의 기본(8)

### 실습 3-4 서식을 사용한 출력의 예 2

```
01 public class Ex03_04 {  
02     public static void main(String[] args) {  
03         System.out.printf("%d / 1 = 2 Wn", 100, 200, 0.5); ----- 정수 2개와 실수  
04         System.out.printf("3 %c Wn", 'a', 'K'); ----- 문자 2개를 출력한다.  
05         System.out.printf("%s %s Wn", "IT CookBook", "JAVA"); ----- 문자열 2개를 출력한다.  
06     }  
07 }
```

3% 8 4% 2 P% 1 365

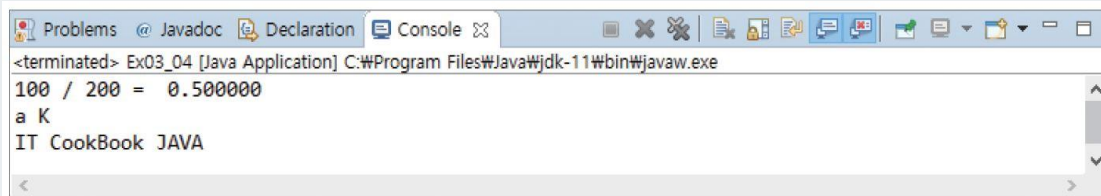


그림 3-6 실행 결과

## 9. System.out.printf() 메소드의 서식 지정(1)

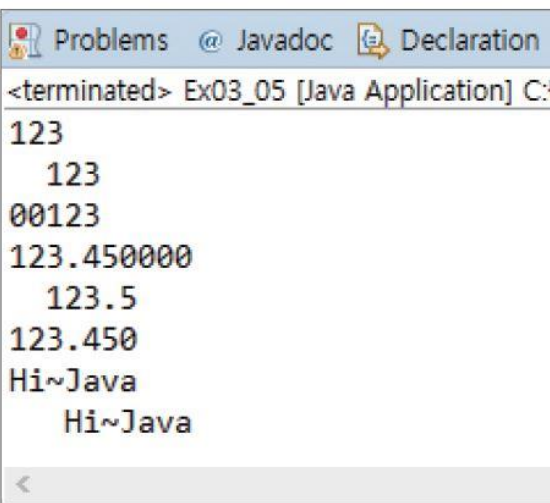
### 실습 3-5 다양한 서식 활용 예 1

```
01 public class Ex03_05 {  
02     public static void main(String[] args) {  
03         System.out.printf("%dwn", 123);  
04         System.out.printf("%5dwn", 123);  
05         System.out.printf("%05dwn", 123);  
06  
07         System.out.printf("%fwn", 123.45);  
08         System.out.printf("%7.1fwn", 123.45);  
09         System.out.printf("%7.3fwn", 123.45);  
10  
11         System.out.printf("%swn", "Hi~Java");  
12         System.out.printf("%10swn", "Hi~Java");  
13     }  
14 }
```

정수형 서식을 활용했다.

실수형 서식을 활용했다.

문자열형 서식을 활용했다.



```
Problems @ Javadoc Declaration  
<terminated> Ex03_05 [Java Application] C:  
123  
 123  
00123  
123.450000  
 123.5  
123.450  
Hi~Java  
  Hi~Java
```

그림 3-7 실행 결과

## 10. System.out.printf() 메소드의 서식 지정(2)

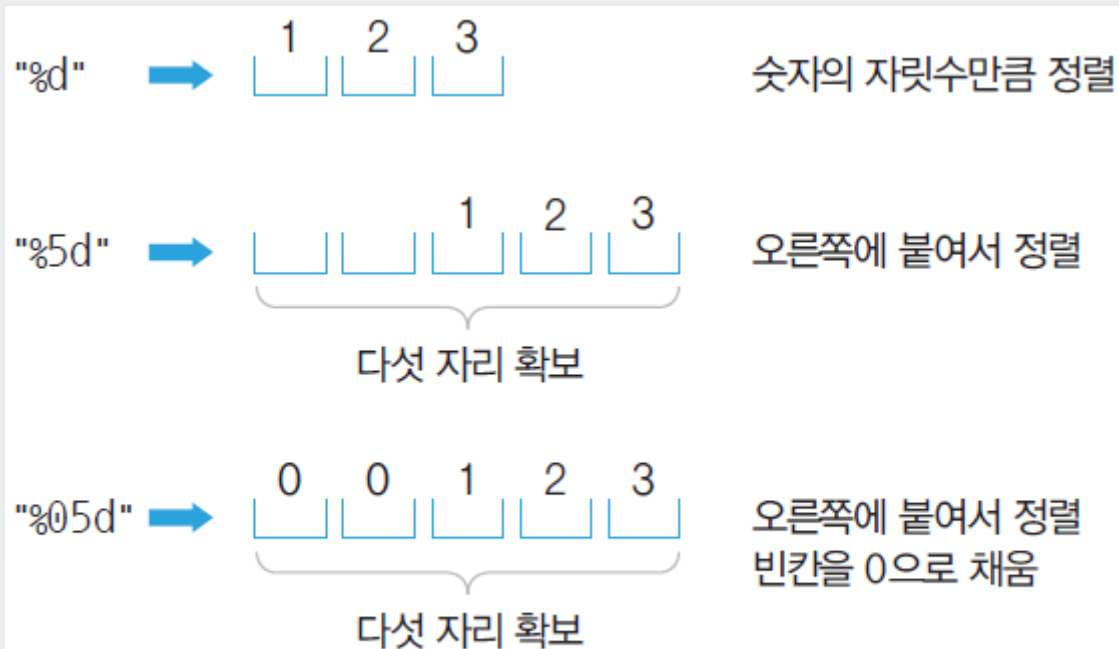


그림 3-8 정수형 데이터 서식 지정

# 11. System.out.printf() 메소드의 서식 지정(3)



그림 3-9 실수형 데이터 서식 지정

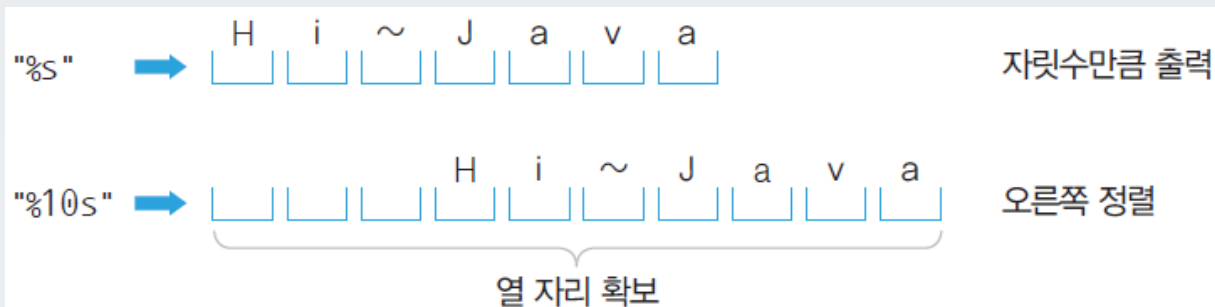


그림 3-10 문자열형 데이터 서식 지정

## 12. System.out.printf() 메소드의 서식 지정(4)

- 다양한 기능의 서식 문자

표 3-2 그 외 다양한 서식 문자

서식 문자	설명	비고
\n	새로운 줄로 이동	 키를 누른 효과
\t	다음 탭으로 이동	 키를 누른 효과
\b	뒤로 한 칸 이동	 키를 누른 효과
\r	줄의 맨 앞으로 이동	 키를 누른 효과
\\	\ 출력	
\'	' 출력	
\"	" 출력	

# 13. System.out.printf() 메소드의 서식 지정(5)

## 실습 3-6 다양한 서식 활용 예 2

```
01 public class Ex03_06 {  
02     public static void main(String[] args) {  
03         System.out.printf("ㄴ줄바꿈ㄴ연습 ㄴ");  
04         System.out.printf("ㄴ탭키ㄴ연습 ㄴ");  
05         System.out.printf("이것을ㄴ뒤어씹니다 ㄴ");  
06         System.out.printf("글자가 ㄴ강조ㄴ되는 효과 ㄴ");  
07         System.out.printf("ㄴㄴㄴ 역슬래시 세개 출력 ㄴ");  
08     }  
09 }
```

다양한 서식 문자를 사용했다.

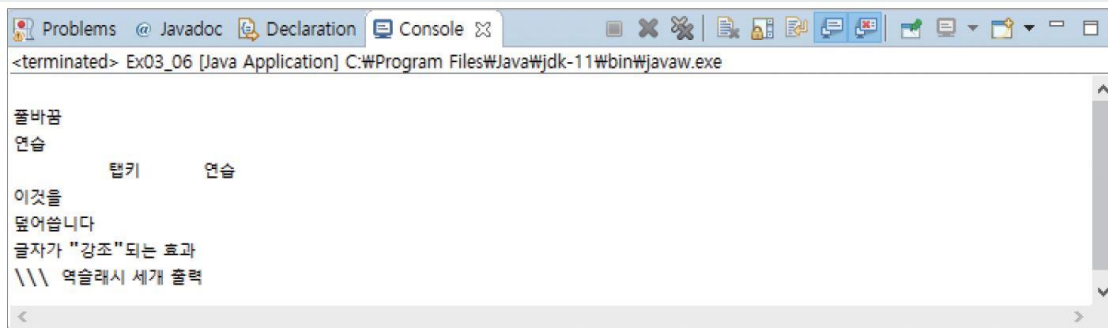
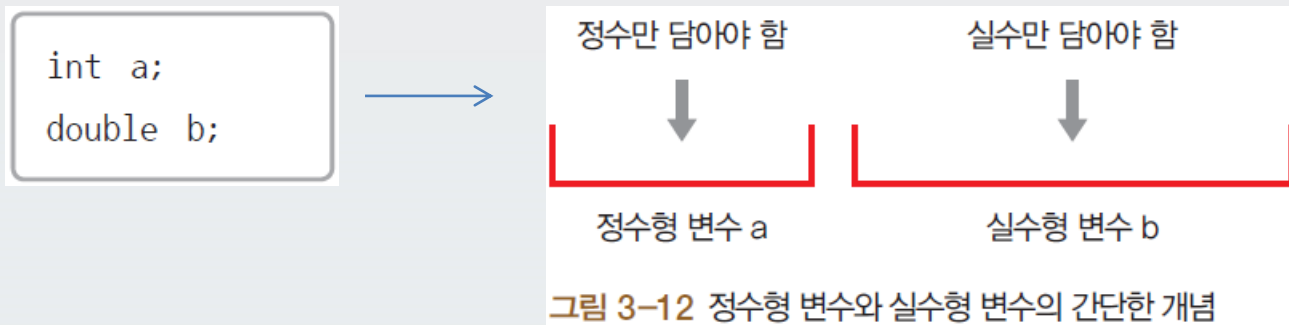


그림 3-11 실행 결과

# 14. 변수(1)

- 변수의 선언
  - 변수 : 어떤 값을 저장하기 위한 메모리 공간



- TIP : 정수형 int는 4byte, 실수형 float는 4byte, 실수형 double은 8byte.  
[그림 3-12]는 실수형이 double이라고 가정한 상태임



## 15. 변수(2)

- 다양한 변수 선언

```
int a;  
int b;
```

==

```
int a, b;
```

① 가능

```
int a;  
float b;  
int c;  
float d;
```

==

② 가능

```
int a, c;  
float b, d;
```

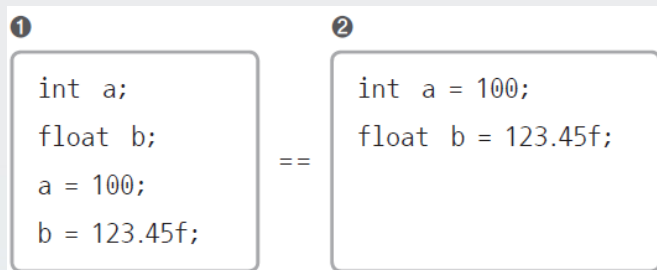
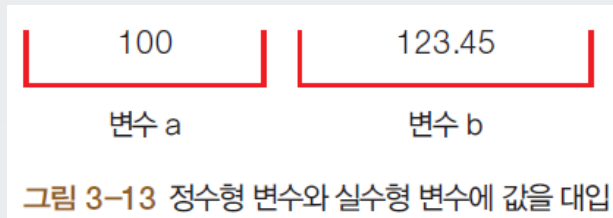
≠

③ 불가능

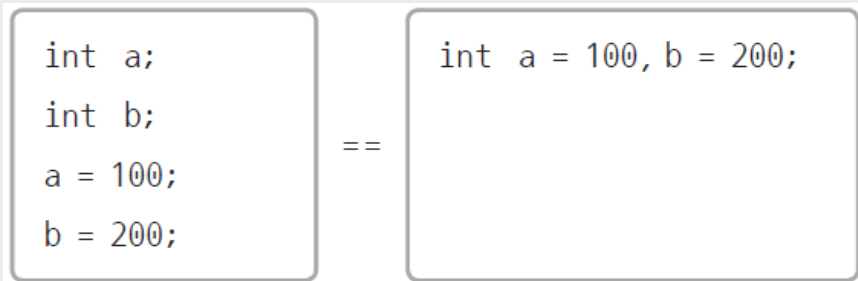
```
int a, float b;  
int c, float d;
```

## 16. 변수(3)

### ■ 변수에 값을 대입하는 방법



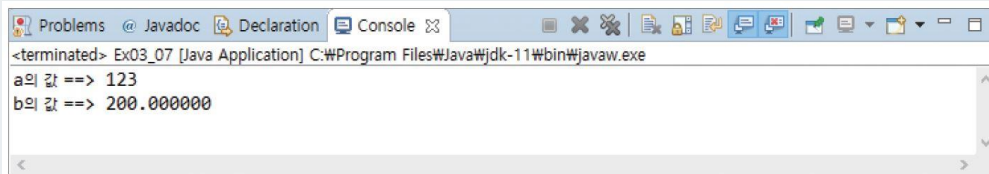
- TIP : 실수형인 float에 값을 대입할 때는 숫자의 맨 뒤에 float를 의미하는 'f'를 붙여야 함
- a, b가 모두 정수형 변수일 때



# 17. 변수(4)

## 실습 3-7 변수에 값을 대입

```
01 public class Ex03_07 {
02     public static void main(String[] args) {
03         int a; ----- 정수형 변수 a를 선언했다.
04         float b; ----- 실수형 변수 b를 선언했다.
05
06         a = (int) 123.45f; ----- 정수형 변수에 실수를 대입했다. → 바람직하지 않다. 오류를 방지하기 위해
                                (int)를 붙여서 정수로 변환했다.
07         b = 200; ----- 실수형 변수에 정수를 대입했다. → 바람직하지 않다.
08
09         System.out.printf("a의 값 ==> %d \n", a);
10         System.out.printf("b의 값 ==> %f \n", b);
11     }
12 }
```



```
<terminated> Ex03_07 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe
a의 값 ==> 123
b의 값 ==> 200.000000
```

그림 3-14 실행 결과

## 18. 변수(5)

- 6행 : 정수형 변수에 실수를 대입할 때의 처리 방식



- 7행 : 실수형 변수에 정수를 대입할 때의 처리 방식



7행을 다음과 같이 수정

```
b = 200.0f;  
또는  
b = (float)200;
```

# 19. 변수(6)

## 실습 3-8 변수에 변수를 대입 1

```
01 public class Ex03_08 {
02     public static void main(String[] args) {
03         int a, b;      —— 정수형 변수가 2개이다.
04         float c, d;    —— 실수형 변수가 2개이다.
05
06         a = 100;       —— a에 정수 100을 대입한다.
07         b = a;         —— b에 a 값을 대입한다.
08
09         c = 111.1f;    —— c에 실수 111.1을 대입한다.
10         d = c;         —— d에 c 값을 대입한다.
11
12         System.out.printf("a, b의 값 ==> %d , %d \n", a, b);
13         System.out.printf("c, d의 값 ==> %5.1f , %5.1f \n", c, d);
14     }
15 }
```

↩ = p [?] :e = q [?] [?] [?]

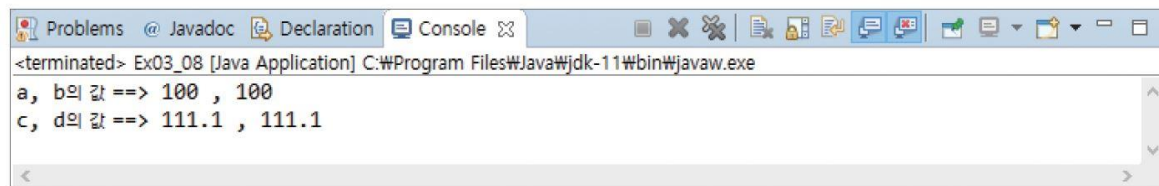
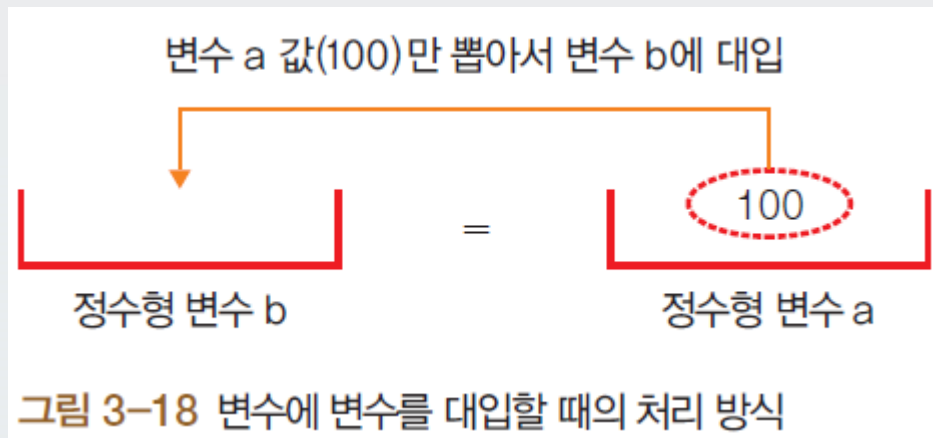


그림 3-17 실행 결과

## 20. 변수(7)

- 6행 : 변수에 변수를 대입할 때의 처리 방식



# 21. 변수(8)

## 실습 3-9 변수에 변수를 대입 2

```
01 public class Ex03_09 {
02     public static void main(String[] args) {
03         int a, b, c, d;
04
05         a = 100 + 100; ----- a에 두 숫자의 계산 결과를 대입한다.
06         b = a + 100; ----- b에 변수와 숫자의 계산 결과를 대입한다.
07         c = a + b - 100; ----- c에 변수의 계산 결과와 숫자의 계산 결과를 대입한다.
08          ----- d에 a, b, c의 덧셈 결과를 대입한다.
09         System.out.printf("a, b, c, d 의 값 ==> %d, %d, %d, %d \n", a, b, c, d);
10
11          ----- a, b, c, d에 모두 같은 값 100을 대입한다(한 문장으로 처리).
12         System.out.printf("a, b, c, d 의 값 ==> %d, %d, %d, %d \n", a, b, c, d);
13
14         a = 100;
15         a = a + 200; ----- 자신의 a 값과 200을 더한 값을 다시 a에 대입한다.
16         System.out.printf("a 의 값 ==> %d \n", a);
17     }
18 }
```

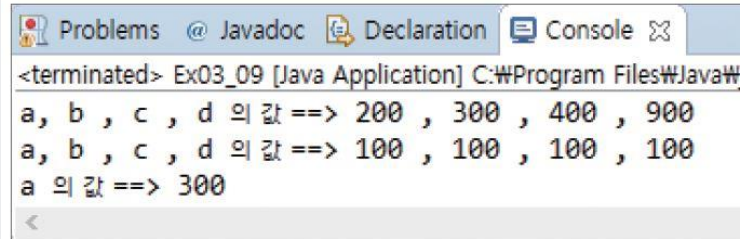
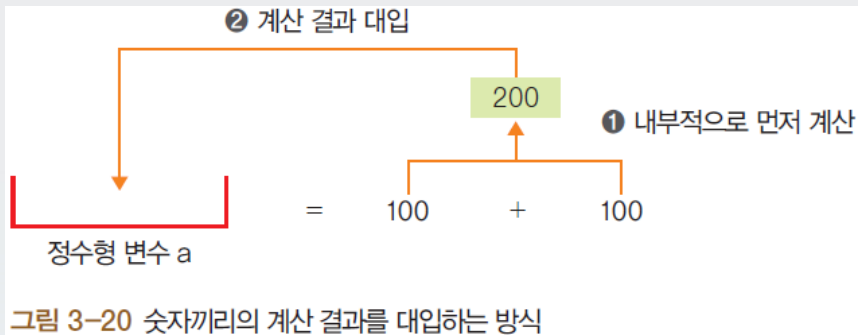


그림 3-19 실행 결과

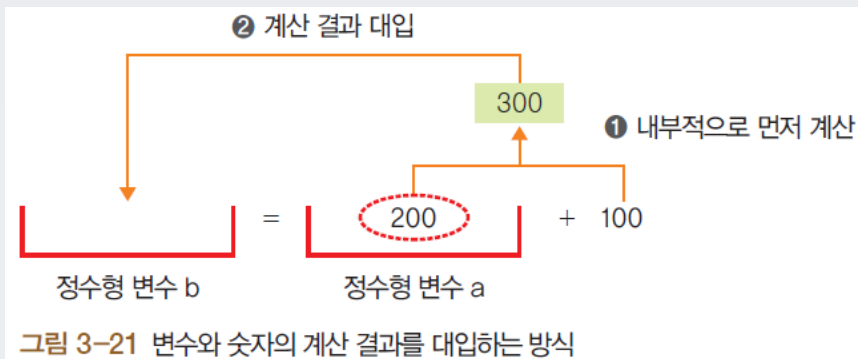
001 = p = 3 = q = e 2 3 + q + e = p 1 3

## 22. 변수(9)

- 5행 : 숫자끼리의 계산 결과를 대입하는 방식



- 6행 : 변수와 숫자의 계산 결과를 대입하는 방식





## 23. 변수(10)

### ■ 11행

```
a = b = c = d = 100;
```

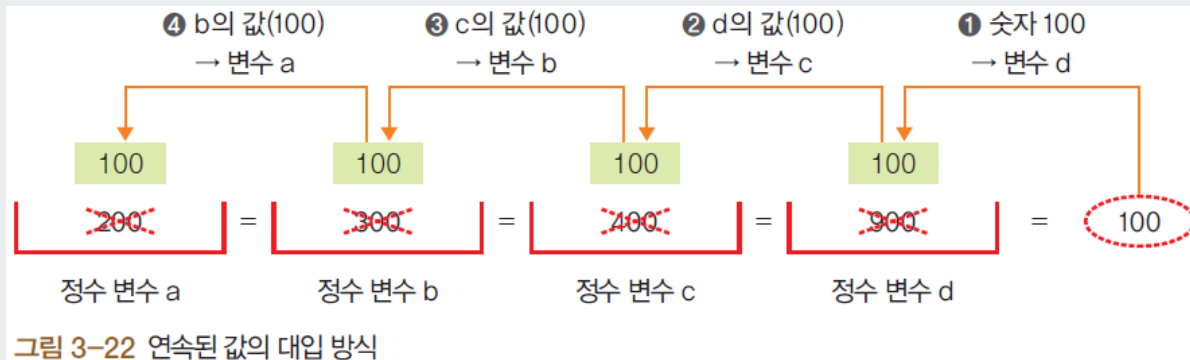
==

```
d = 100;
```

```
c = d;
```

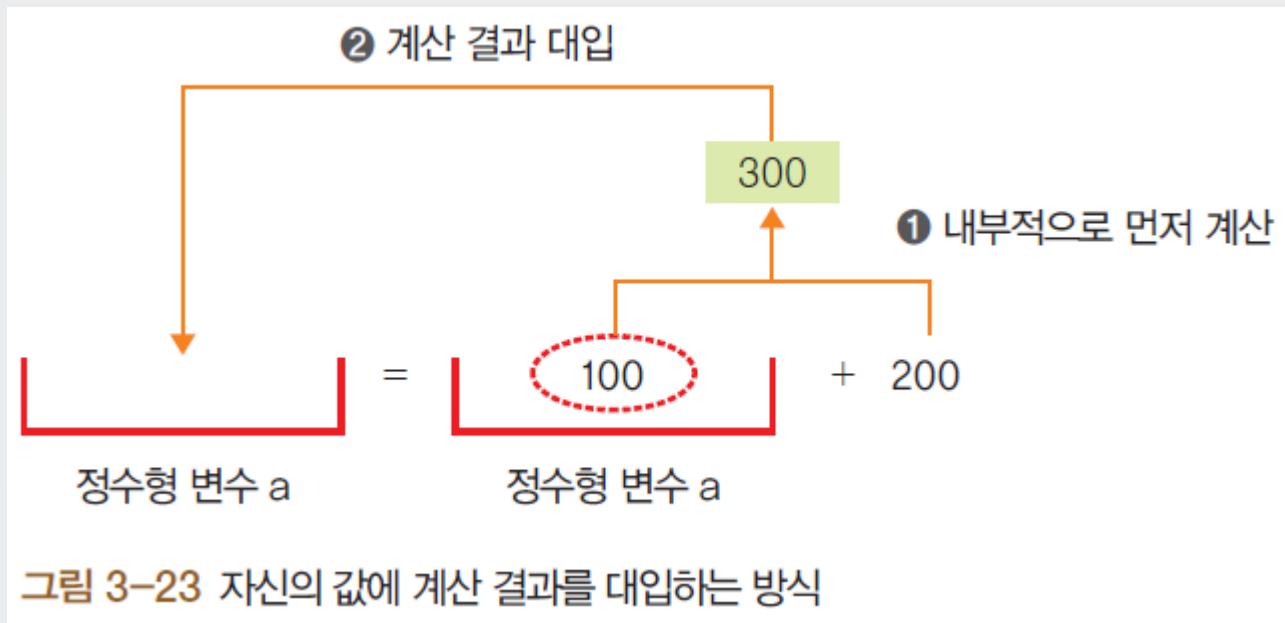
```
b = c;
```

```
a = b;
```



## 24. 변수(11)

- 15행 : 자신의 값에 계산 결과를 대입하는 방식



## 25. 변수(12)

### ■ 대입 연산자와 변수의 위치

- 대입 연산자(=)를 사용하면 오른쪽의 것이 왼쪽에 대입됨
- 대입 연산자(=)의 오른쪽에는 상수(숫자), 변수, 계산 값이 모두 올 수 있음



(X)



(O)

## 26. 데이터 형식과 배열(1)

### ■ 비트

- 0(OFF)과 1(ON)만 존재

전기 스위치		의미	2진수	10진수
		꺼짐, 꺼짐	00	0
		꺼짐, 켜짐	01	1
		켜짐, 꺼짐	10	2
		켜짐, 켜짐	11	3

그림 3-26 2개의 전기 스위치와 2진수, 10진수의 비교

- $n$ 개의 전기 스위치로 표현할 수 있는 가짓 수 =  $2^n$
- 3비트로 표현할 수 있는 가짓수는  $2^3=8$ 개, 4비트로 표현할 수 있는 가짓수는  $2^4=16$ 개

## 27. 데이터 형식과 배열(2)

### ■ 진수

표 3-3 10진수, 2진수, 16진수의 변환

10진수(0~9)	2진수(0, 1)	16진수(0~F)
00	0000	0
01	0001	1
02	0010	2
03	0011	3
04	0100	4
05	0101	5
06	0110	6
07	0111	7
08	1000	8
09	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

**TIP/** 보기 좋게 하기 위해 10진수 0은 00으로, 2진수 0은 0000으로 자릿수를 맞춰서 나타냈다. 또한 8진수도 사용할 수 있는데 활용도가 비교적 낮다.

## 28. 데이터 형식과 배열(3)

### ■ 바이트

- JAVA에서 가장 많이 사용되는 단위. 바이트는 8개의 비트가 합쳐진 것

표 3-4 비트와 바이트의 크기에 따른 숫자의 범위

비트 수	바이트 수	표현 개수	2진수	10진수	16진수
1		$2^1=2$	0~1	0~1	0~1
2		$2^2=4$	0~11	0~3	0~3
4		$2^4=16$	0~1111	0~15	0~F
8	1	$2^8=256$	0~11111111	0~255	0~FF
16	2	$2^{16}=65536$	0~11111111 11111111	0~65535	0~FFFF
32	4	$2^{32}$ =약 42억	0~...	0~약 42억	0~FFFF FFFF
64	8	$2^{64}$ =약 1800경	0~...	0~약 1800경	0~...

## 29. 데이터 형식과 배열(4)

### ■ 2진수를 10진수로 변환하는 방법

2진수	1	0	0	1		0	0	1	1
	×	×	×	×		×	×	×	×
	$2^7$	$2^6$	$2^5$	$2^4$		$2^3$	$2^2$	$2^1$	$2^0$
	128	0	0	16		0	0	2	1
	+								
10진수	147								

그림 3-27 2진수를 10진수로 변환하는 방법

## 30. 데이터 형식과 배열(5)

- 2진수를 16진수로 변환한 후 10진수로 변환하는 방법

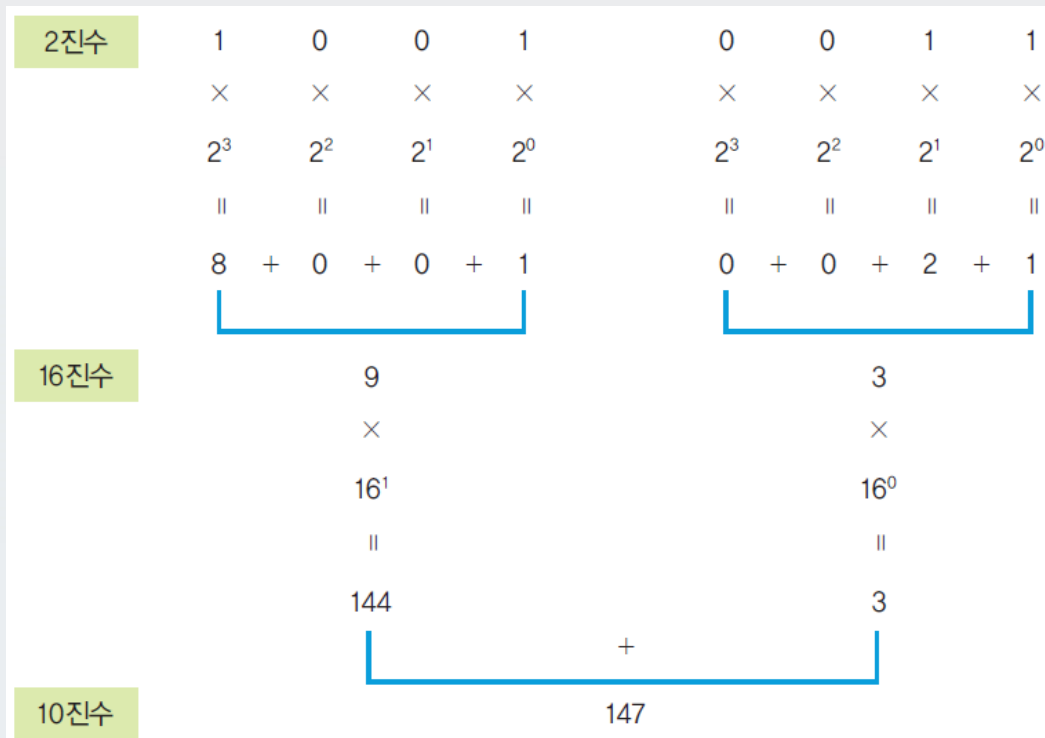
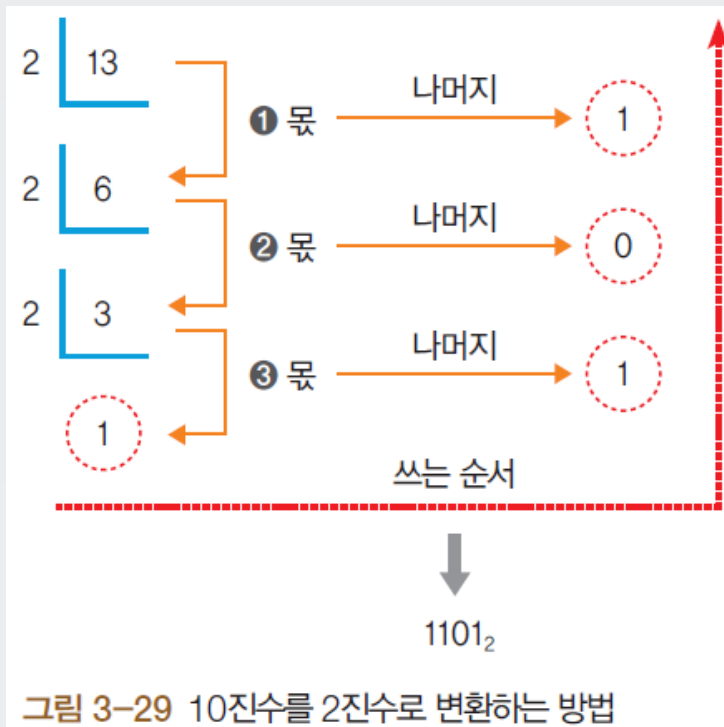


그림 3-28 2진수를 16진수로 변환한 후 10진수로 변환하는 방법



# 31. 데이터 형식과 배열(6)

- 2진수 변환 연습
  - 10진수를 2진수로 변환



## 32. 데이터 형식과 배열(7)

- 16진수를 2진수로 변환. 16진수라는 것을 나타내기 위해  $13_{16}$ 과 같이 표현함

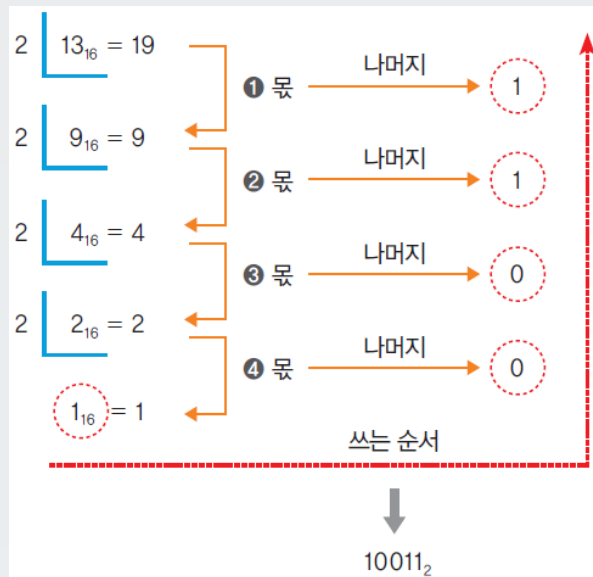


그림 3-30 16진수를 2진수로 변환하는 방법

- JAVA에서 16진수를 표현할 때는 숫자 앞에 '0x' 또는 '0X'를 붙이면 된다. 예를 들어 'a=10'은 a에 10진수 10을 대입하라는 것이지만, 'a=0x10'은 16진수 10(일영이라 읽으며 10진수로는 16이다)을 대입하라는 의미

## 33. 데이터 형식과 배열(8)

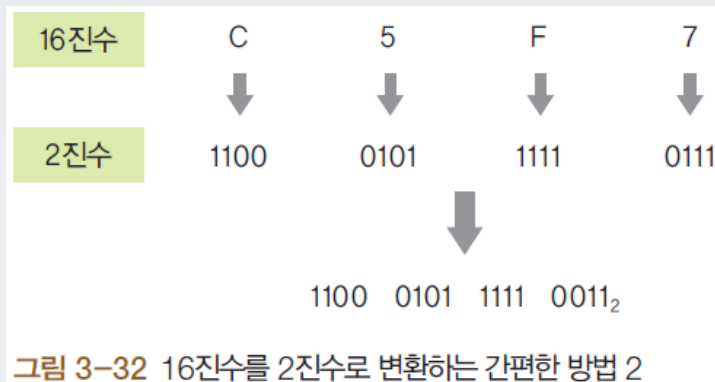
### ■ 16진수와 2진수 변환표

표 3-5 16진수와 2진수 변환표

16진수	2진수	16진수	2진수
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

## 34. 데이터 형식과 배열(9)

- 16진수를 2진수로 변환하는 간편한 방법



## 35. 데이터 형식과 배열(10)

### ■ 소수점이 없는 정수형

표 3-6 정수형 데이터 형식

정수형 데이터 형식	의미	크기	값의 범위
byte	아주 작은 정수형	1바이트	$-2^7(-128) \sim 2^7-1(127)$
short	작은 정수형	2바이트	$-2^{15}(-32768) \sim 2^{15}-1(32767)$
int	정수형	4바이트	$-2^{31}(\text{약 } -21\text{억}) \sim 2^{31}-1(\text{약 } 21\text{억})$
long	큰 정수형	8바이트	$-2^{63}(\text{약 } -900\text{경}) \sim 2^{63}-1(\text{약 } 900\text{경})$

- JAVA는 정수형 중에서 int형을 기본 정수형으로 취급함

## 36. 데이터 형식과 배열(11)

### 실습 3-10 소수점이 없는 정수형

```
01 public class Ex03_10 {  
02     public static void main(String[] args) {  
03         int a=100, b=200; ----- 정수형 변수 a와 b에 값을 지정한다.  
04         float result; ----- 실수형 변수 result를 선언한다.  
05  
06         result = a / b; ----- a를 b로 나눈 다음 결과를 실수형 변수 result에 대입한다(0.5를 예상).  
07  
08         System.out.printf ("%f  \n", result);  
09     }  
10 }
```

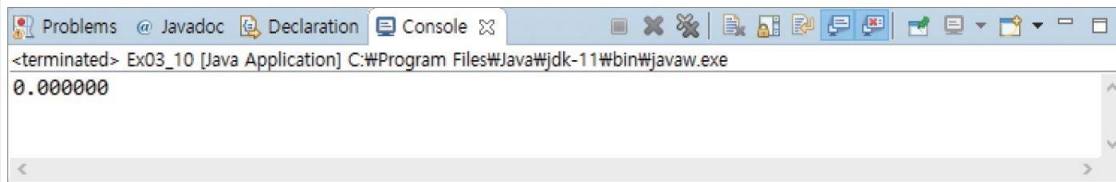
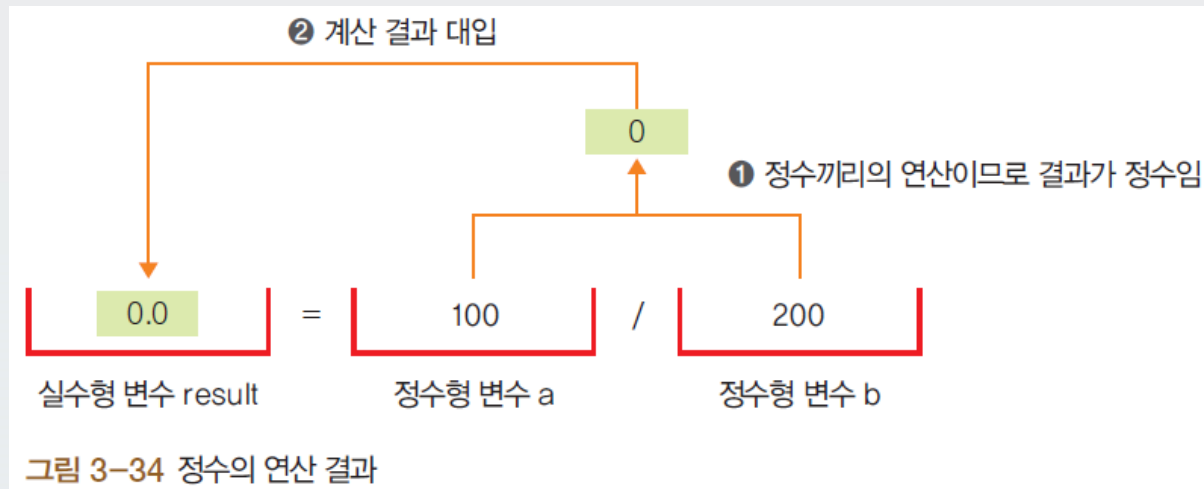


그림 3-33 실행 결과

## 37. 데이터 형식과 배열(12)

### 정수의 연산 결과



정수 +, -, \*, / 정수 = 정수

정수 +, -, \*, / 실수 = 실수

실수 +, -, \*, / 실수 = 실수

## 38. 데이터 형식과 배열(13)

### ■ 소수점이 있는 실수형

표 3-7 실수형 데이터 형식

실수형 데이터 형식	의미	크기	값의 범위
float	실수형	4바이트	약 $-3.4 \times 10^{38} \sim 3.4 \times 10^{38}$
double	큰 실수형	8바이트	약 $-1.79 \times 10^{308} \sim 1.79 \times 10^{308}$

- JAVA는 실수형 중에서 double형을 기본 실수형으로 취급.  
그래서 3.14라고 쓰면 double형 값으로 인식하고, 이 값을 float형으로 만들려면 숫자 뒤에 float를 의미하는 'f'를 붙여야 하므로 3.14f가 됨



## 39. 데이터 형식과 배열(14)

### 실습 3-11 소수점이 있는 실수형

```
01 public class Ex03_11 {  
02     public static void main(String[] args) {  
03         float a = 0.1234567890123456789012345f; ----- float형 변수 a에 정밀도 25자리를 입력했다.  
04         double b = 0.1234567890123456789012345; ----- double형 변수 b에 정밀도 25자리를 입력했다.  
05  
06         System.out.printf("%30.25f %n", a);  
07         System.out.printf("%30.25f %n", b);  
08     }  
09 }
```



```
<terminated> Ex03_11 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe  
0.12345679104328156000000000  
0.12345678901234568000000000
```

그림 3-35 실행 결과

## 40. 데이터 형식과 배열(15)

### ■ 문자형 데이터 형식 - 아스키코드와 유니코드

- 아스키코드(ASCII) : 컴퓨터에서 표현하는 문자(특히 키보드에 있는 영문, 기호, 숫자 등)를 0~127에 대응시킨 코드

표 3-8 아스키코드

아스키코드	10진수	16진수
0~9	48~57	0x30~0x39
A~Z	65~90	0x41~0x5A
a~z	97~122	0x61~0x7A

정수형 97을 문자형으로 표현하면 a가 됨

```
char ch = 'a';
```

==

```
char ch = 97 ;
```

# 41. 데이터 형식과 배열(16)

- 한 글자를 표현하는 문자형 - 문자형은 문자 또는 기호 하나를 저장하는 형식

표 3-9 문자형 데이터 형식

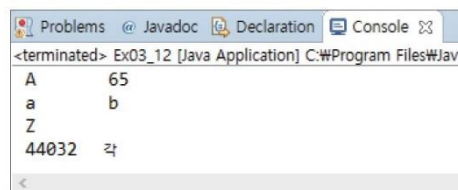
문자형 데이터 형식	의미	크기	값의 범위
char	문자형 또는 정수형	2바이트	$0 \sim 2^{16}-1$ (65535)

- char형에는 문자뿐만 아니라 값의 범위에 해당하는 정수를 대입할 수 있음.  
char형을 2바이트 크기의 정수형으로 취급해도 상관없다는 의미.  
즉 char형의 크기가 2바이트(516비트)이므로 표현할 수 있는 글자 수는 65536가지이고,  
값의 범위는 0~65535. 따라서 아스키코드표의 0~127을 비롯해 한글, 중국어, 아랍어 등을  
모두 표현할 수 있음

## 42. 데이터 형식과 배열(17)

실습 3-12 문자형 변수 사용 예 1

```
01 public class Ex03_12 {
02     public static void main(String[] args) {
03         char a, b, c, d, e; ----- 문자형 변수 5개를 선언한다.
04
05         a = 'A'; ----- 문자형 변수 a에 'A'를 대입한다.
06         System.out.printf(" %c %t", a);
07         System.out.printf(" %d %n", (int) a); ----- 문자형 변수 a를 문자형과 정수형으로 출력한다.
08
09         b = 'a'; ----- 문자형 변수 b에 'a'를 대입한다.
10         c = (char) (b + 1); ----- 문자형 변수 b에 1를 더하여 문자형 변수 c에 대입한다.
                                결과는 char로 캐스팅한다.
11         System.out.printf(" %c %t", b);
12         System.out.printf(" %c %n", c);
13
14         d = 90; ----- 문자형 변수 d에 숫자 90을 대입한다.
15         System.out.printf(" %c %n", d);
16
17         d = '가'; ----- 문자형 변수 d에 한글을 대입한다.
18         e = (char) (d + 1); ----- 문자형 변수 d에 1를 더하여 문자형 변수 e에 대입한다.
                                결과는 char로 캐스팅한다.
19         System.out.printf(" %d %t", (int) d);
20         System.out.printf(" %c %n", e);
21     }
22 }
```

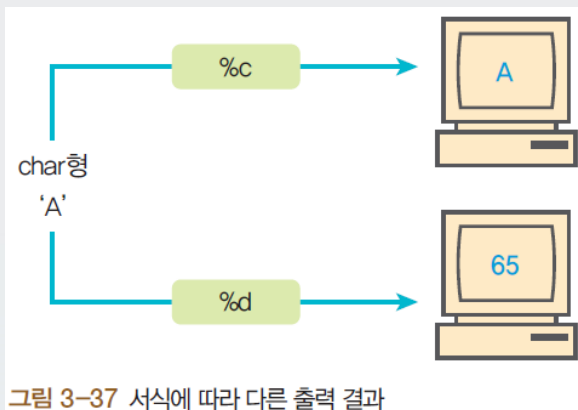


```
<terminated> Ex03_12 [Java Application] C:\Program Files\Java
A      65
a      b
Z
44032  각
```

그림 3-36 실행 결과

## 43. 데이터 형식과 배열(18)



- 3~7행



- 9행 : 변수 b에 문자 'a'를 대입, 10행에서 변수 c에 '변수 b의 값11'을 대입. 그런데 변수 b에는 문자 'a'가 들어 있지만 이는 숫자 97과 동일하므로, 여기에 1를 더한 값인 98이 변수 b에 들어감. 이를 12행에서 문자형으로 출력하면 문자 'b'가 됨
- 14행: d가 문자형 변수이지만 숫자 90일수도 있음. 숫자 90은 문자 'Z'와 동일하므로 결국 변수 d에 'Z'를 입력한 것과 같음. 그 결과 'Z'가 출력
- 17행 : 변수 d에 한글 '가'를 대입, 19행에서 정수로 출력하니 한글 '가'는 44032 코드 값인 것이 확인. 18행에서 변수 d에 1을 증가시키고 20행에서 출력하여 '각'이 출력

## 44. 데이터 형식과 배열(19)

### 실습 3-13 문자형 변수 사용 예 2

```
01 public class Ex03_13 {
02     public static void main(String[] args) {
03         int a, b;
04         char c, d;
05
06         a = 0xF90A;
07         b = 0x50;
08
09          ----- 정수형 변수 a를 문자형으로 출력한다.
10
11         c = (char) b; ----- 문자형 변수 c에 정수형 변수 b 값을 대입한다.
12          ----- 문자형 변수 c를 문자형으로 출력한다.
13
14         d = '#'; ----- 문자형 변수 d에 '#'를 대입한다.
15         System.out.printf("%c의 ASCII값은 %d 입니다 %n", d, (int) d); ----- 문자형 변수 d를 두
16     }                                           가지 형태로 출력한다.
17 }
```

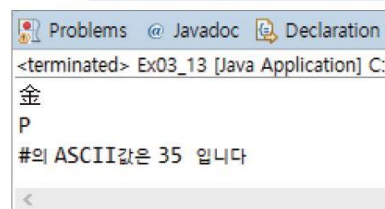


그림 3-38 실행 결과

## 45. 데이터 형식과 배열(20)

### ■ 참과 거짓을 표현하는 불(Boolean)

표 3-10 문자형 데이터 형식 boolean

문자형 데이터 형식	의미	크기	값의 범위
boolean	참, 거짓을 저장	1바이트	true, false만 저장

- 실제로 참을 1, 거짓을 0으로 취급하면 1바이트(=8비트)가 아닌 1비트만으로도 표현이 가능하지만, JAVA는 최소 접근 단위가 1바이트이므로 크기는 1바이트를 사용

## 46. 데이터 형식과 배열(21)

### 실습 3-14 불형 사용 예

```
01 public class Ex03_14 {  
02     public static void main(String[] args) {  
03         boolean boo1, boo2; ----- 불형 변수 2개를 선언한다.  
04  
05         boo1 = true; ----- boo1 변수에 true를 대입한다.  
06         System.out.printf("%s Wn", boo1); ----- boo1 변수를 출력한다.  
07  
08         boo2 = (10 == 20); ----- boo2 변수에 10과 20을 비교한 결과를 대입한다.  
09         System.out.printf("%s Wn", boo2); ----- boo2 변수를 출력한다.  
10     }  
11 }
```

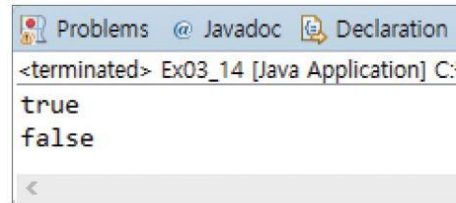


그림 3-39 실행 결과

8행 : (10 == 20)은 '10과 20이 같다'의 결과이므로 true 또는 false가 반환



## 47. 데이터 형식과 배열(22)

### ■ 여러 글자를 표현하는 문자열

- 문자열의 개념은 ‘문자형의 집합’이고 ‘문자열’이라는 데이터 형식은 따로 존재하지 않음  
문자형 데이터 형식 char는 한 글자만 저장
- 문자열은 데이터 형식으로 지원되지 않고 String 클래스로 사용할 수 있음

표 3-11 문자형 데이터 형식 string

문자형 데이터 형식	의미	크기	값의 범위
String	문자열	(입력한 글자 수 × 2) 바이트	저장되는 모든 값은 글자로 저장됨

## 48. 데이터 형식과 배열(23)

실습 3-15 문자열 사용 예

```
01 public class Ex03_15 {
02     public static void main(String[] args) {
03         String str1 = "IT CookBook 입니다.";
04         String str2 = "10";
05         String str3 = "20";
06
07         str1 = "Java 입니다. ";
08
09         System.out.printf("%s Wn", str1);
10         System.out.printf("%s Wn", str2 + str3);
11     }
12 }
```

----- str1 문자열 변수를 선언하고, 선언과 동시에 값을 입력한다.

----- str2, str3 문자열 변수를 선언하고 각각에 문자열 "10"과 "20"을 대입한다.

----- str1 문자열 변수에 값을 대입한다.

----- str1을 출력한다.

----- str2와 str3의 합계를 출력한다.

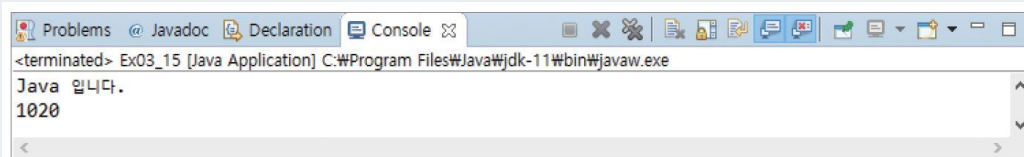


그림 3-40 실행 결과



다음 시간

# 소프트웨어 품질



송실사이버대학교

송실사이버대학교의 강의콘텐츠는  
저작권법에 의하여 보호를 받는바, 무단  
전재, 배포, 전송, 대여 등을 금합니다.

\*사용서체 : 나눔글꼴