

# 컴퓨터 프로그래밍 (Computer Programming)

이 선 순



### 3. 변수와 데이터 형식



# 목차

1. `System.out`의 기본
2. `System.out.printf()` 메소드의 서식 지정
3. 변수
4. 데이터 형식

04

데이터 형식

(문자형, 불형, 이스케이프문자)

## 04. 데이터 형식

### ■ 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 ;
```

## 04. 데이터 형식

### ■ 문자형 데이터 형식 - 한 글자를 표현하는 문자형 : char형식

- 한 글자를 표현하는 문자형은 문자 또는 기호 하나를 저장하는 형식

'A' '한'

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(16비트)이므로 표현할 수 있는 글자 수는 65536가지이고, 값의 범위는 0~65535. 따라서 아스키코드표의 0~127을 비롯해 한글, 중국어, 아랍어 등을 모두 표현할 수 있음

## 04. 데이터 형식

### ■ 문자형 데이터 형식 - 한 글자를 표현하는 문자형 : char형식

- 작은 따옴표로 감싼 문자 리터럴은 유니코드로 변환되어 저장

=> char 타입은 정수 타입

```
char var1 = 'A';    //유니코드: 65  
char var2 = 'B';    //유니코드: 66  
char var3 = '가';    //유니코드: 44032  
char var4 = '각';    //유니코드: 44033
```

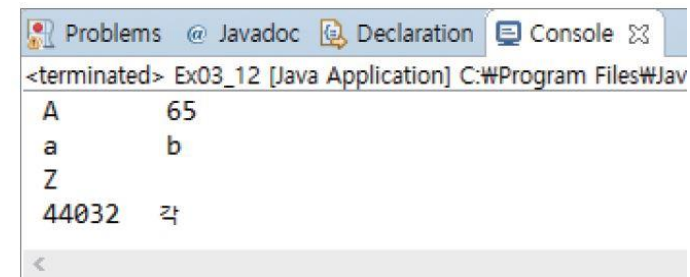
- char는 정수 타입이므로 10 진수 또는 16진수 형태의 유니코드 저장 가능

```
char c = 65;        //10진수  
char c = 0x0041;    //16진수
```

## 04. 데이터 형식

### 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\Jav
A      65
a      b
Z
44032  각
```

그림 3-36 실행 결과



## 04. 데이터 형식

### ■ 문자형 데이터 형식 - 한 글자를 표현하는 문자형

#### ■ 3~7행

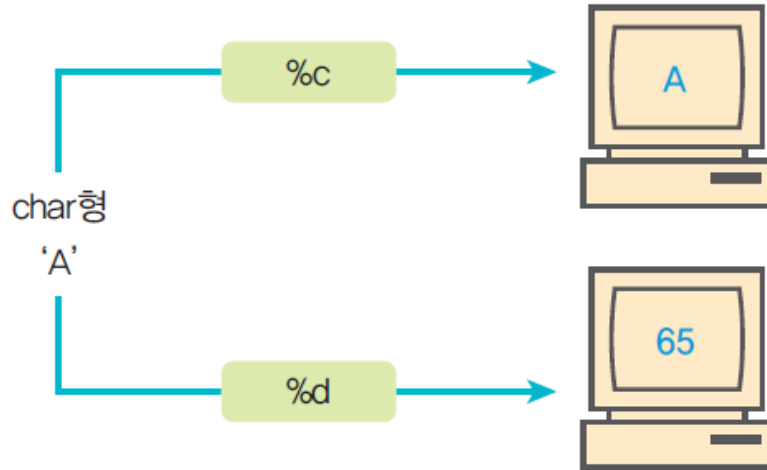


그림 3-37 서식에 따라 다른 출력 결과

- 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

## 04. 데이터 형식

### 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         1 ----- 정수형 변수 a를 문자형으로 출력한다.
10
11         c = (char) b; ----- 문자형 변수 c에 정수형 변수 b 값을 대입한다.
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 }
```

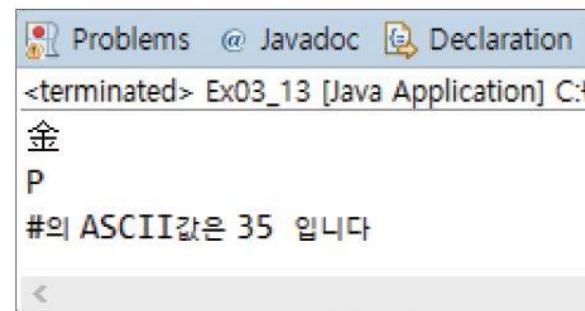


그림 3-38 실행 결과

1 System.out.printf("%c \n", a); 2 System.out.printf("%c \n", c);

## 04. 데이터 형식

### ■ 올바른 문자 표현

- 문자는 반드시 ' '로 묶어야 하고 한 글자만 올 수 있다. 다음은 모두 틀린 표현이다.

```
char a;  
a = 'Ab';  
a = "A";  
a = "Ab";
```

## 04. 데이터 형식

### ■ char 형식

```
1 package sec02.exam04;
2
3 public class CharExample {
4     public static void main(String[] args) {
5         char c1 = 'A';           //문자를 직접 저장
6         char c2 = 65;             //십진수로 저장
7         char c3 = '\u0041';       //16진수로 저장
8
9         char c4 = '가';           //문자를 직접 저장
10        char c5 = 44032;          //십진수로 저장
11        char c6 = '\uac00';       //16진수로 저장
12
13        System.out.println(c1);
14        System.out.println(c2);
15        System.out.println(c3);
16        System.out.println(c4);
17        System.out.println(c5);
18        System.out.println(c6);
19    }
20 }
21
22
23
```

Problems @ Javadoc Declaration Console

<terminated> CharExample [Java Application] C:\Program Files\Java\jdk-11.0.6\bin

A  
A  
A  
가  
가  
가

## 04. 데이터 형식

### ■ 문자형 데이터 형식 - 여러 글자를 표현하는 문자열 : String 형식

- 여러 글자를 표현함
- 문자열의 개념은 '문자형의 집합'이고 '문자열'이라는 데이터 형식은 따로 존재하지 않음
- 문자형 데이터 형식 char는 한 글자만 저장
- 큰 따옴표로 감싼 문자들을 문자열이라고 함
- 문자열은 char형식에 저장할 수 없음.
- 문자열은 String 타입 변수로 저장함 => 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

```
char var1 = "A";  
char var2 = "홍길동";
```

```
String var1 = "A";  
String var2 = "홍길동";
```

## 04. 데이터 형식

### ■ String 타입 변수

```
1 package sec02.exam05;
2
3 public class StringExample {
4     public static void main(String[] args) {
5         String name = "봉준호";
6         String job = "영화감독";
7         System.out.println(name);
8         System.out.println(job);
9     }
10 }
11
```

Problems @ Javadoc Declaration Console

<terminated> StringExample [Java Application] C:\Program Files\Java\j

봉준호  
영화감독

## 04. 데이터 형식

### 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 \n", str1);
10         System.out.printf("%s \n", str2 + str3);
11     }
12 }
```

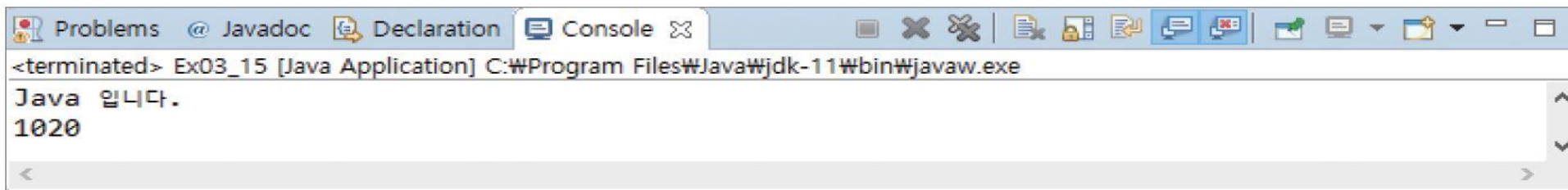
----- str1 문자열 변수를 선언하고, 선언과 동시에 값을 입력한다.

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

----- str1 문자열 변수에 값을 대입한다.

----- str1을 출력한다.

----- str2와 str3의 합계를 출력한다.



```
<terminated> Ex03_15 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe
Java 입니다.
1020
```

그림 3-40 실행 결과

## 04. 데이터 형식

### ■ 문자형 데이터 형식 - 참과 거짓을 표현하는 불 : Boolean형식

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
- 참과 거짓에 해당하는 true와 false 리터럴을 저장하는 타입
- 두 가지 상태값에 따라 조건문과 제어문의 실행 흐름을 변경하는데 사용

```
boolean stop = true;  
boolean state = false;
```



## 04. 데이터 형식

### ■ 문자형 데이터 형식 참과 거짓을 표현하는 불(Boole)

#### 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 }
```

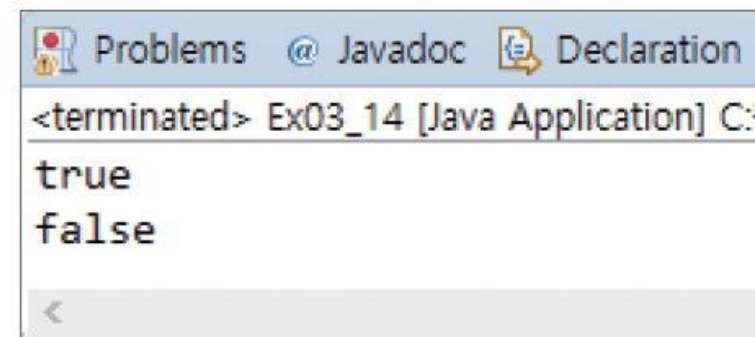


그림 3-39 실행 결과

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

## 04 데이터형식

```
1 package sec02.exam08;
2
3 public class BooleanExample {
4     public static void main(String[] args) {
5         boolean stop = true;
6         if(stop) {
7             System.out.println("중지합니다.");
8         } else {
9             System.out.println("시작합니다.");
10        }
11    }
12 }
```

Problems @ Javadoc Declaration Console

<terminated> BooleanExample [Java Application] C:\Program Files  
중지합니다.

## 04. 데이터 형식

### ■ 이스케이프 문자

- 문자열 내부에 \는 이스케이프 문자를 뜻함
- 이스케이프 문자를 사용하면 특정 문자를 포함시키거나, 문자열의 출력을 제어할 수 있음
- 예. 문자열 내부에 " 문자 포함

```
String str = "나는 \"자바\"를 좋아합니다.";
System.out.println(str);
```

→ 나는 "자바"를 좋아합니다.

## 04. 데이터 형식

### ■ 이스케이프 문자

- 예. 문자열 출력 제어

```
String str = "번호\t이름\t나이";  
System.out.println(str);
```

→ 번호      이름      나이  
    └──┬──┘ └──┬──┘  
      tab 공간    tab 공간

```
String str = "홍길동\n감자바";  
System.out.println(str);
```

→ 홍길동  
    감자바

# 04 데이터 형식

## ■ 이스케이프 문자

이스케이프 문자	출력 용도
\t	탭만큼 띄움
\n	줄 바꿈(라인 피드)
\r	캐리지리턴
\"	" 출력
'	' 출력
\\	\ 출력
\u16진수	16진수 유니코드에 해당하는 문자 출력

## 04 데이터 형식

### ■ 이스케이프 문자 출력

```
1 package sec02.exam06;
2
3 public class EscapeExample {
4     public static void main(String[] args) {
5         System.out.println("번호\\t이름\\t직업 ");
6         System.out.print("행 단위 출력\\n");
7         System.out.print("행 단위 출력\\n");
8         System.out.println("우리는 \\\"개발자\\\" 입니다.");
9         System.out.print("봄\\t여름\\t가을\\t겨울");
10    }
11 }
12
13
```

Problems Javadoc Declaration Console

<terminated> EscapeExample [Java Application] C:\\Program Files\\Ja

번호 이름 직업  
행 단위 출력  
행 단위 출력  
우리는 "개발자" 입니다.  
봄여름가을겨울

질문은 이메일을 이용해주세요.  
[ds.june2@gmail.com](mailto:ds.june2@gmail.com)

**감사합니다**