

제1장 – 기본적인 프로그램

Outline

- 1.1 Hello Word!
- 1.2 변수
- 1.3 수치형
- 1.4 문자와 문자열
- 1.5 배열
- 1.6 다차원 배열(1)
- 1.7 다차원 배열(2)



1.1 Hello World!

- 프로그램 만들기

예 >>

```
class 클래스명 부분부터 시작합니다.  
Class Print {  
    public static void main(String [] args) {  
        System.out.println("Hello World !") ;  
    }  
}
```

문자열을 표시합니다.

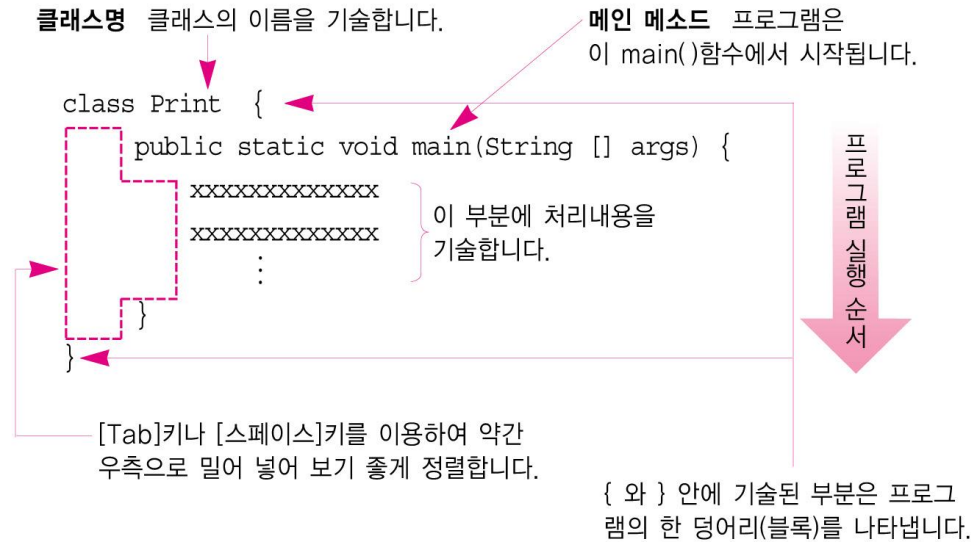
실행결과 >>

Hello World!
■



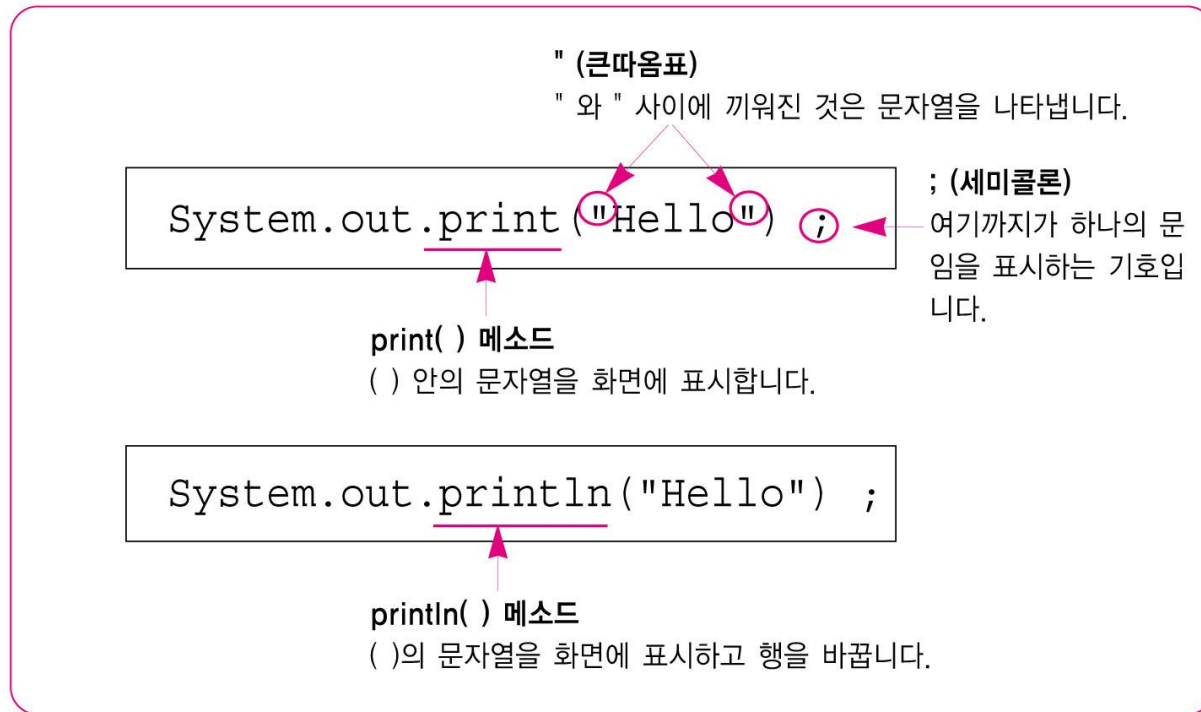
1.1 Hello World!

• 프로그램 만들기



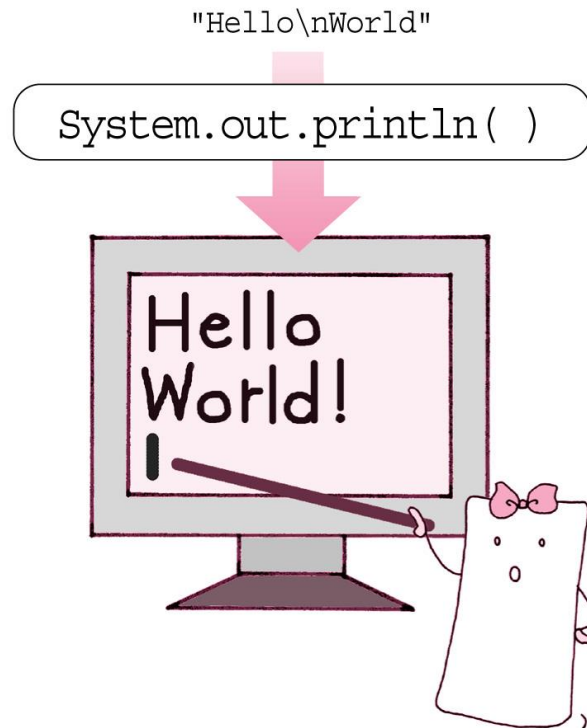
1.1 Hello World!

- 문자열 표시하기



1.1 Hello World!

- 문자열 표시하기

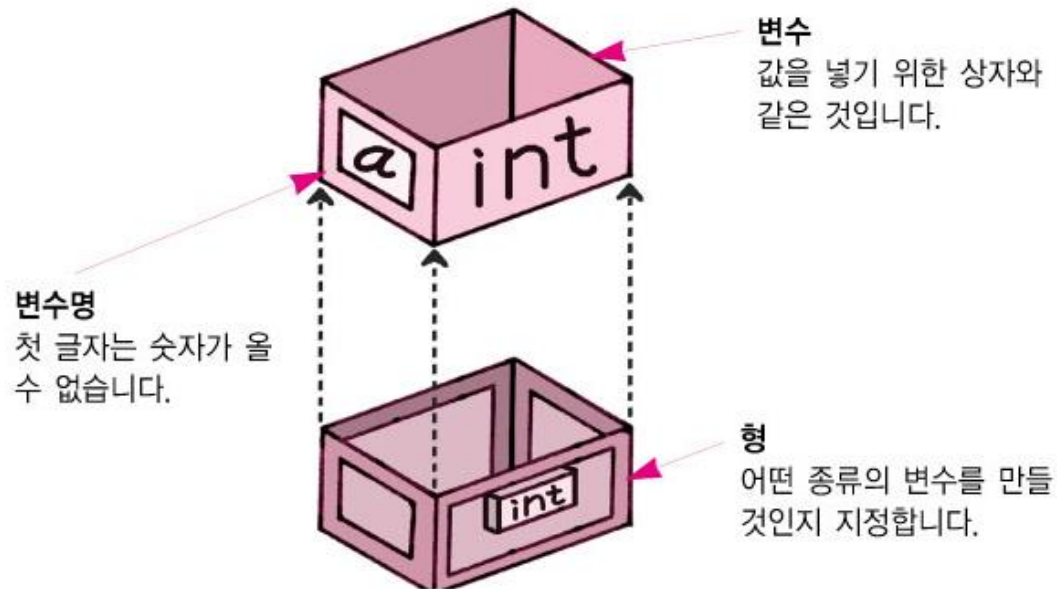


1.2 변수

- 변수의 선언과 대입

```
int a ;
```

정수(integer, 인티저) 값이 들어가는 a라는 이름의 변수를 준비합니다. 이것을 'int형 변수 a를 선언한다'고 합니다.



1.2 변수

• 변수의 선언과 대입

문은 ;로 구분되며, 한 줄로 늘어 쓰는 것도 가능합니다.

<pre>a = 2 ; b = 3 ;</pre>	→	<pre>a = 2 ; b = 3 ;</pre>
----------------------------	---	----------------------------

변수의 선언과 수치의 대입을 다음과 같이 정리해 볼 수 있습니다.

두 변수의 선언

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



```
int a, b ;
```

한 변수 선언과 대입

```
int a ; a = 2 ;
```



```
int a = 2 ;
```

두 변수 선언과 대입

```
int a ; int b ;
a = 2 ; b = 3 ;
```



```
int a = 2, b = 3 ;
```

1.2 변수

- 변수의 선언과 대입

```
a = 2;
```

int 형으로 만들어진 변수 a에 2라는 값을 넣습니다.
이것을 '변수 a에 2를 **대입**한다'고 합니다.



1.2 변수

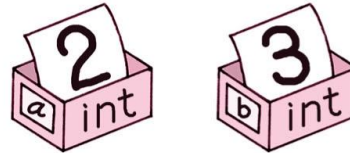
• 변수를 표시하기

```
class Data {
    public static void main (String [] args) {
        int a;
        int b;
        a = 2 ;
        b = 3 ;
        a = b ;

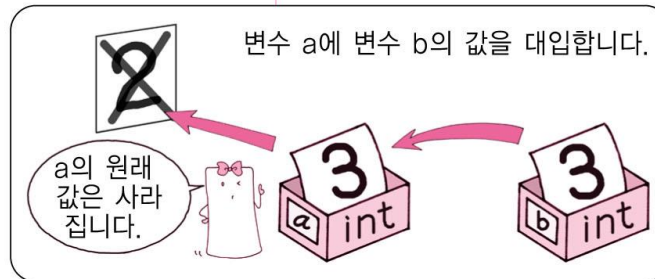
        System.out.println(a) ;
    }
}
```

<< 예

변수 a, b를 선언하고, 각각 2와 3을 대입합니다.



변수 a에 변수 b의 값을 대입합니다.



3

a의 값을 표시한다

<< 실행결과

■



1.3 수치형

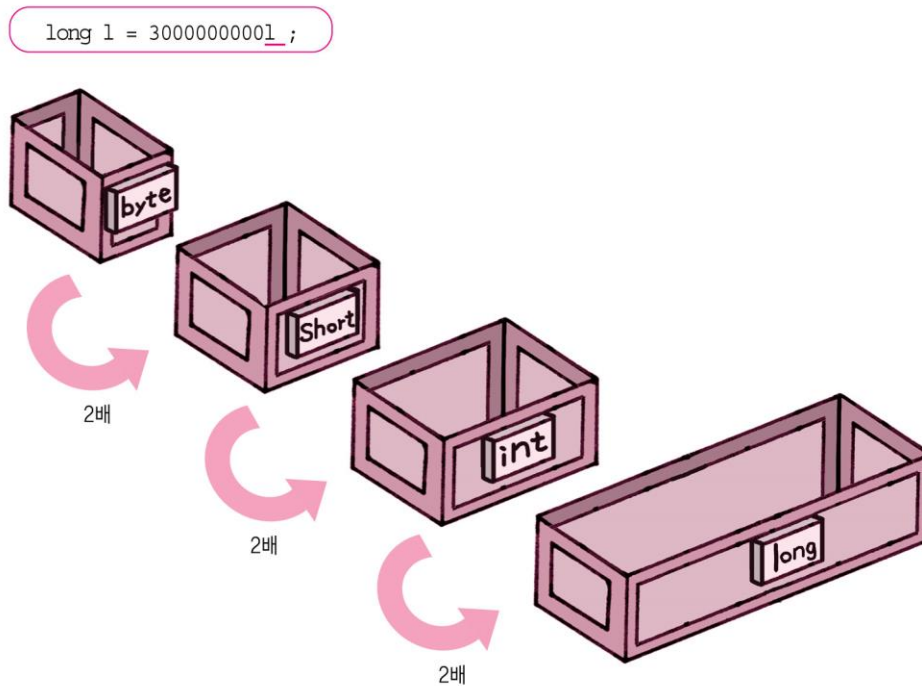
- 정수형

형의 이름	읽는 법	들어가는 값의 범위	사이즈(비트 수)
byte	바이트	-128 ~ 127	8
short	쇼트	-32768~32767	16
int	인트	-2147483648~ 2147483647	32
long	롱	-9223372036854775808~ 9223372036854775807	64



1.3 수치형

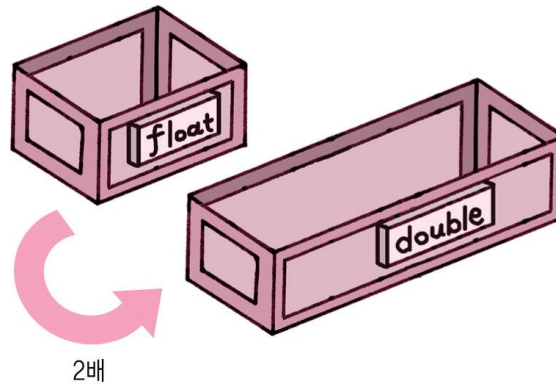
- 정수형



1.3 수치형

- 실수형

형의 이름	읽는 법	들어가는 값의 대략적인 범위	사이즈(비트 수)
float	플로트	$\pm 3.4 \times 10^{38} \sim \pm 1.4 \times 10^{-45}$	32
double	더블	$\pm 1.8 \times 10^{308} \sim \pm 4.9 \times 10^{-324}$	64



1.3 수치형

- 실수형

```
float f = 3.4f ;
```

<< 예

```
class Data {
    public static void main (String [] args) {
        정수형 <-- byte a = 127 ; <-- 정수값
        <-- short b = 10000 ; <-- 정수값
        실수형 <-- float c = 1.2f ; <-- 실수값
        <-- double d = 0.34 ; <-- 실수값
        변수의 선언과 대입
        (초기화)

        System.out.println(a) ;
        System.out.println(b) ;
        System.out.println(c) ;
        System.out.println(d) ;
        처리
    }
}
```

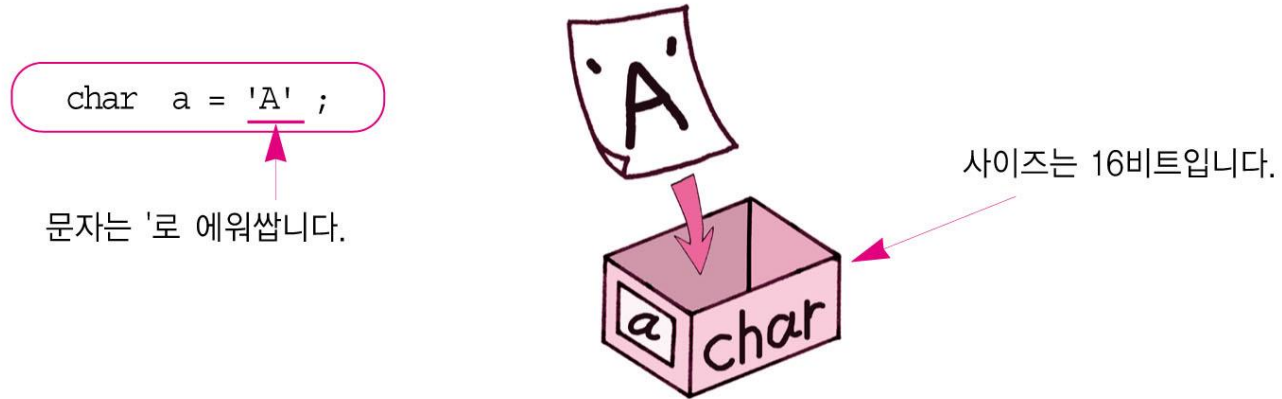
127
10000
1.2
0.34
■

<< 실행결과



1.4 문자와 문자열

- 문자

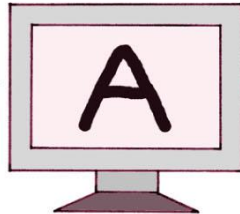


1.4 문자와 문자열

- 문자의 표시

문자 A를 그대로 지정하여 표시한다

```
System.out.println('A') ;
```



문자 A를 문자코드로 지정하여 표시한다

```
System.out.println('₩u0041') ;
```



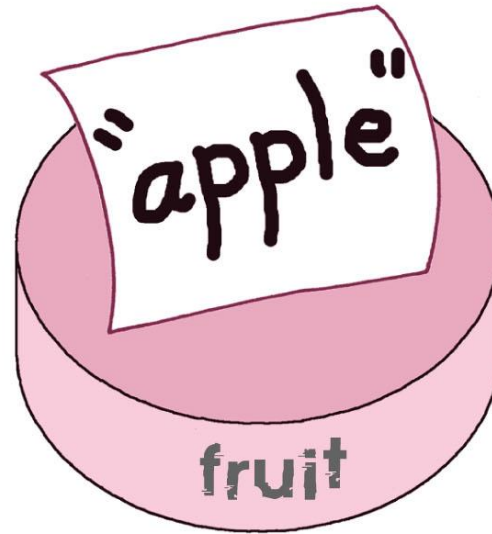
*Unicode의 ₩u0041은 A입니다.

1.4 문자와 문자열

- 문자열

```
String fruit = "apple" ;
```

문자열은 "로 에워쌉니다.



1.4 문자와 문자열

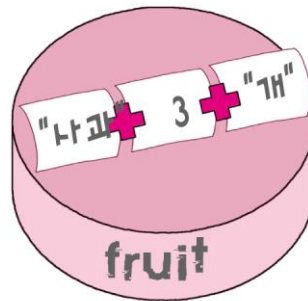
- 문자열의 결합

```
class Fruit {  
    public static void main (String[] args) {  
        String apple = "사과" ;  
        int a = 3 ;  
  
        String fruit = apple + a + "개" ;  
        System.out.println(fruit) ;  
    }  
}
```

<< 예

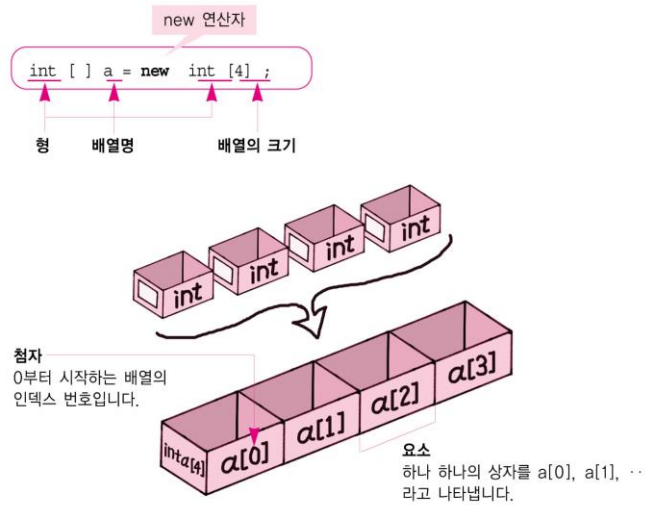
사과3개
■

<< 실행결과



1.5 배열

- 배열의 개념



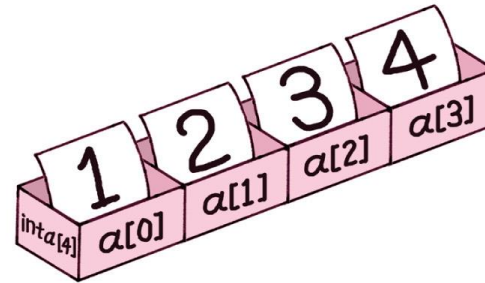
```
int [ ] a = new int [ ] { 1, 2, 3, 4 } ;
```

{ } 안에 데이터가 몇 개 있느냐로 자동적으로
요소수가 결정됩니다.

1.5 배열

- 배열의 개념

```
int [] a = { 1, 2, 3, 4 } ;
```



1-5 배열

- 배열 요소의 참조와 대입

```
int [] a = new int [4] ;  
int  n  = 1 ;  
a[0] = 1 ;  
a[1] = 2 ;  
a[2] = 3 ;  
a[3] = 4 ;  
System.out.println(a[n]) ;
```

← a[0]~ a[3]의 값을 대입

← a[1]의 값을 표시

✗

```
int [ ] a = { 1, 2, 3, 4 } ;  
System.out.println(a[4]) ;
```



1-5 배열

- 배열 요소의 참조와 대입

```
class Number {  
    public static void main( String [] args ) {  
        int [] a = { 1, 2, 3, 4 } ;  
        System.out.println(a[0]) ;  
        System.out.println(a[1]) ;  
        System.out.println(a[2]) ;  
        System.out.println(a[3]) ;  
    }  
}
```

<< 예

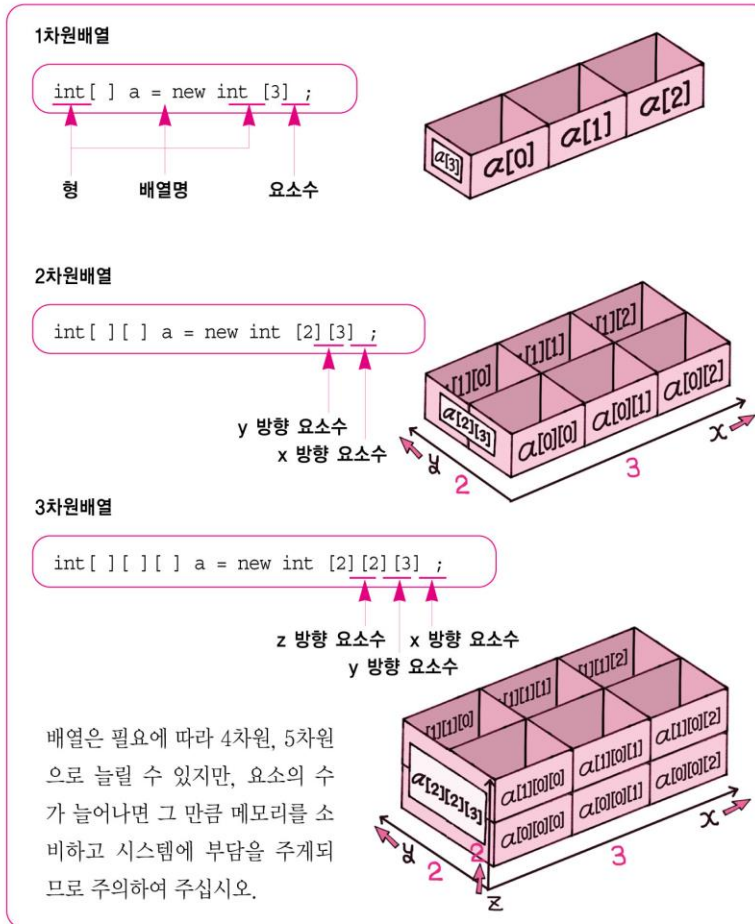
1
2
3
4
■

<< 실행결과



1.6 다차원 배열(1)

- 다차원 배열이란?



1.6 다차원 배열(1)

- 다차원 배열에 대한 대입, 초기화, 참조

```
int a [2] [3] = {  
    { 10, 20, 30 } ,  
    { 40, 50, 60 }  
} ;
```

```
a[0] [2] = 0 ;
```

```
System.out.println( a[1] [0] );
```

초기화

{ }와 콤마의 조합에 주의

a[0][2]에 0을 대입

a[1][0]을 참조



1.6 다차원 배열(1)

- 다차원 배열에 대한 대입, 초기화, 참조

```
class Number {  
    public static void main(String[] args) {  
        int [][] a = {  
            { 10, 20 },  
            { 30, 40 },  
            { 50, 60 }  
        } ;  
  
        System.out.println("a[0][0]=" + a[0][0]  
                           + " a[0][1]=" + a[0][1]) ;  
        System.out.println("a[1][0]=" + a[1][0]  
                           + " a[1][1]=" + a[1][1]) ;  
        System.out.println("a[2][0]=" + a[2][0]  
                           + " a[2][1]=" + a[2][1]) ;  
    }  
}
```

<< 예

```
a[0][0]=10 a[0][1]=20  
a[1][0]=30 a[1][1]=40  
a[2][0]=50 a[2][1]=60
```

■

<< 실행결과



1.7 다차원 배열(2)

- 다차원 배열의 요소 수

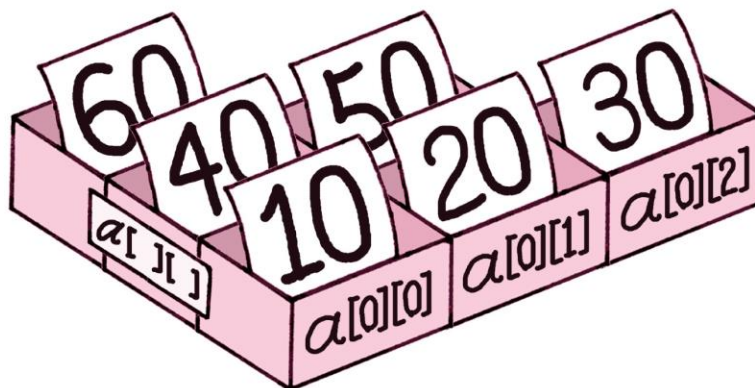
```
int [][] a = {  
    { 10, 20, 30 },  
    { 40, 50 } ,  
    { 60 }  
};
```

배열	값
a[0][0]	10
a[0][1]	20
a[0][2]	30
a[1][0]	40
a[1][1]	50
a[2][0]	60



1.7 다차원 배열(2)

- 다차원 배열의 요소 수

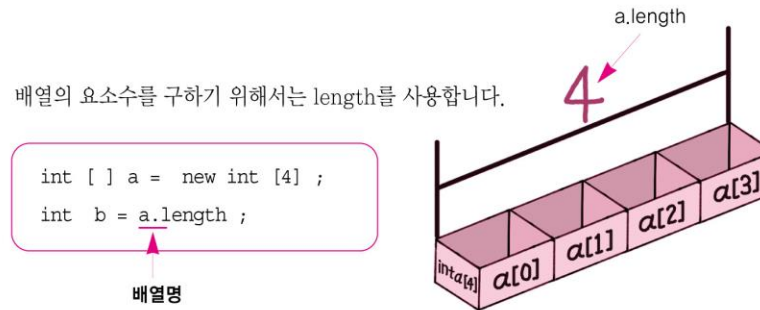


선언만을 할 경우는 다음과 같이 기술합니다.

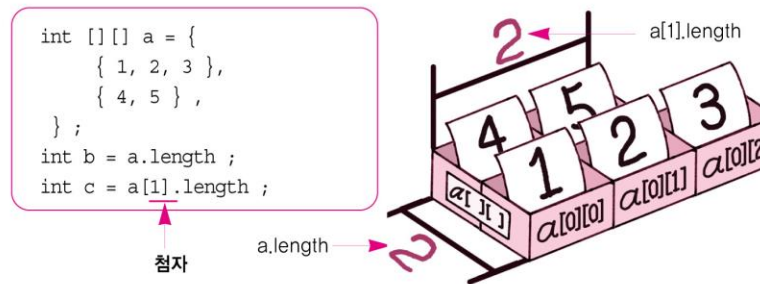
```
int [][] a = new int [3] [] ;  
a[0] = new int [3] ;  
a[1] = new int [2] ;  
a[2] = new int [1] ;
```

1.7 다차원 배열(2)

- 배열의 요소를 구한다



다차원 배열의 요소를 구하기 위해서는 다음과 같이 기술합니다.



1.7 다차원 배열(2)

- 배열의 요소를 구한다

```
class Number2 {  
    public static void main(String[] args) {  
        int [][] a = {  
            { 10, 20, 30 },  
            { 40, 50 },  
            { 60, 70, 80, 90 }  
        } ;  
  
        System.out.println("일차원 배열의 요소수 " + a.length) ;  
        System.out.println("a[0]의 요소수 " + a[0].length) ;  
        System.out.println("a[1]의 요소수 " + a[1].length) ;  
        System.out.println("a[2]의 요소수 " + a[2].length) ;  
    }  
}
```

<< 예

일차원 배열의 요소수 3
a[0]의 요소수 3
a[1]의 요소수 2
a[2]의 요소수 4

■

<< 실행결과

