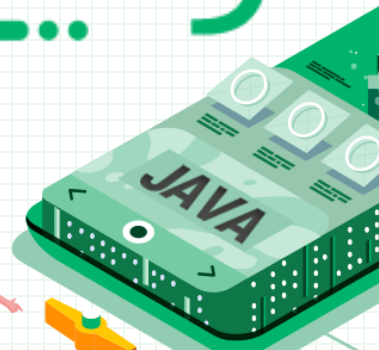




# 안드로이드 프로그래밍을 위한 자바기초 \_...



## ⑤ 배열의 기본 문법 이해하기



## 학습목표

- 배열을 이해하고 프로그래밍에 적용할 수 있다.
- 배열을 활용하여 프로그래밍을 할 수 있다.
- 다차원 배열을 이해하고 프로그래밍에 적용할 수 있다.



## 학습내용

- 배열 이해하기
- 배열 활용하기
- 다차원 배열 이해하기

## ⑤ 배열의 기본 문법 이해하기

## 배열 이해하기

## ➤ 배열(array)의 개념

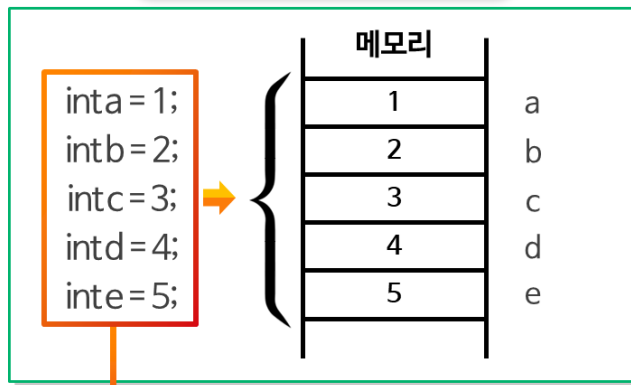
## 1) 배열이란?

- 동일한 자료형 여러 개를 묶어서 처리하는 자료구조
- 변수이름과 인덱스(index)를 이용하여 데이터를 표현

## 2) 배열의 필요성

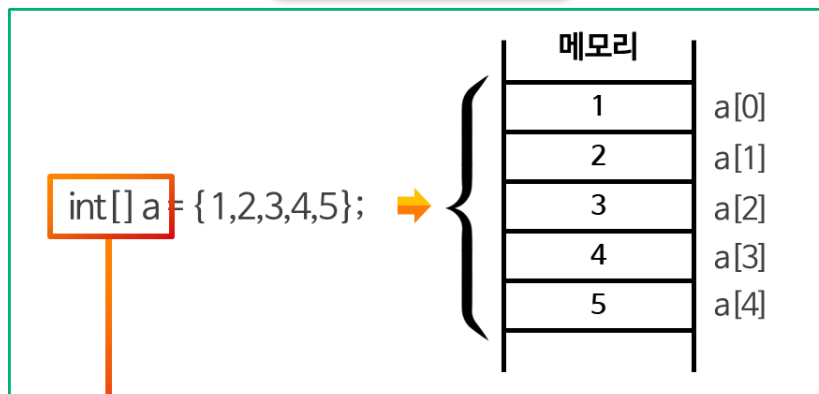
- 동일한 자료형 여러 개를 선언하는 경우 변수가 여러 개 필요
- 배열을 이용하면 하나의 변수만 필요하여 효율적인 프로그래밍 가능

## 배열을 사용하지 않는 경우



5개의변수가필요

## 배열을 사용하는 경우



1개의변수가필요

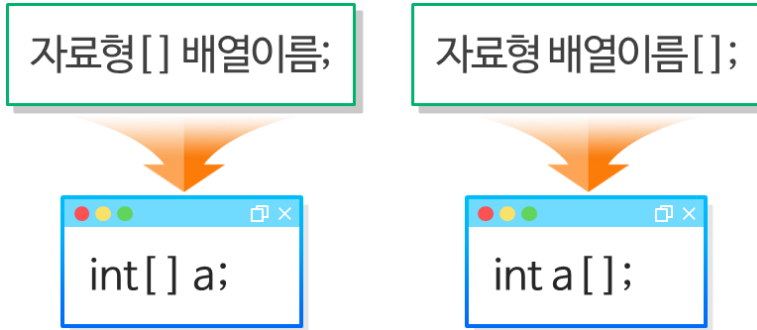
## ⑤ 배열의 기본 문법 이해하기

## 배열 이해하기

### ▶ 배열 선언 및 생성하기

#### 1) 배열 선언 방법

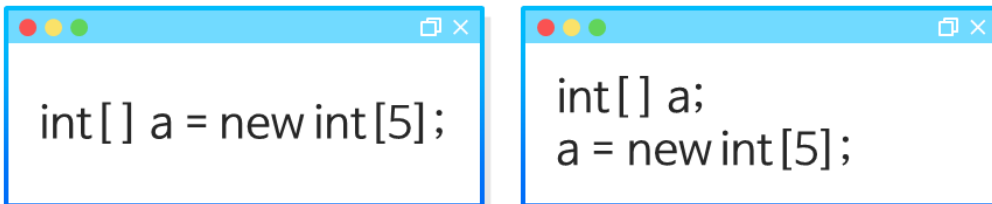
- 배열 선언 시 배열 요소의 개수는 지정하지 않음



#### 2) 배열 생성 방법

- (1) 배열을 생성하는 것은 메모리를 할당하는 작업임
- (2) new 연산자를 이용하여 배열에 메모리를 할당하면 배열이 생성됨
- (3) 배열 생성 시 배열 요소의 개수를 지정함

자료형[] 배열이름 = new 자료형[배열 요소의 개수];



## ⑤ 배열의 기본 문법 이해하기

## 배열 이해하기

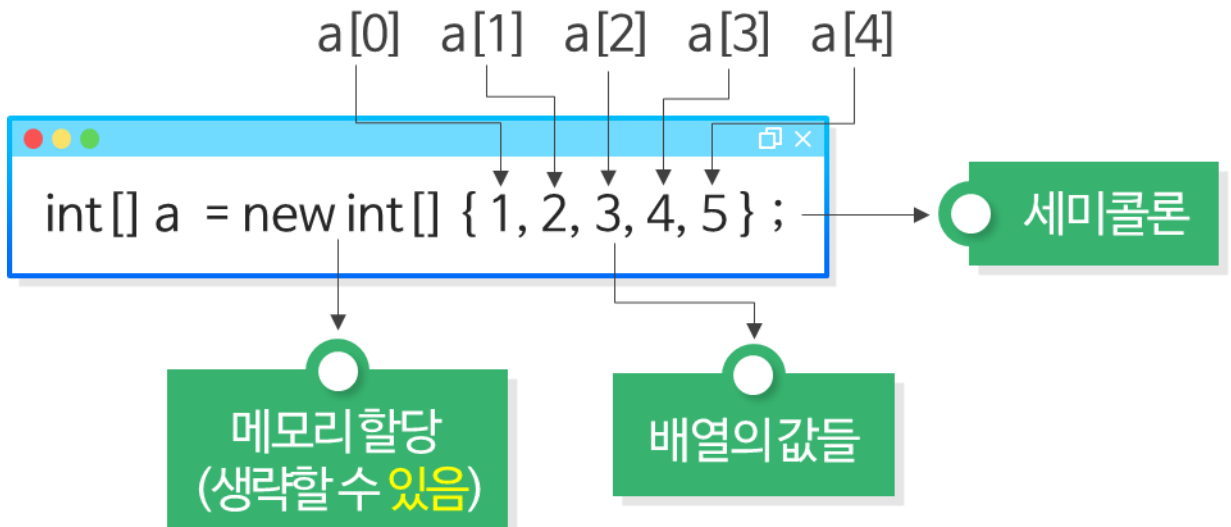
## 3) 배열 초기화

(1) 배열 선언 및 생성 후 배열 요소의 값들을 지정하는 방법

```
int[] a = new int[5];
a[0] = 1;
a[1] = 2;
a[2] = 3;
a[3] = 4;
a[4] = 5;
```

- ✓ 배열 요소의 개수
- ✓ 배열 인덱스 0부터 사용

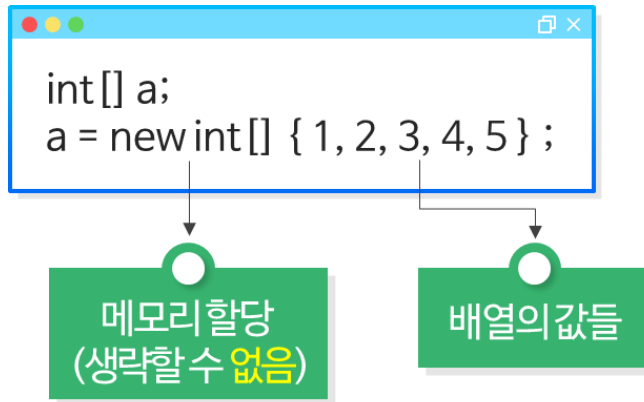
(2) 배열 선언과 동시에 배열 요소의 값들을 지정하는 방법



## ⑤ 배열의 기본 문법 이해하기

## 배열 이해하기

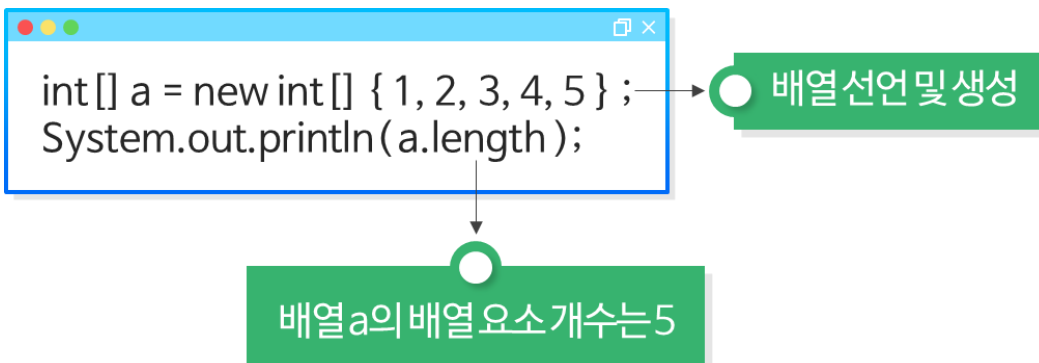
(3) 배열 선언 후 배열 요소의 값들을 지정하는 방법



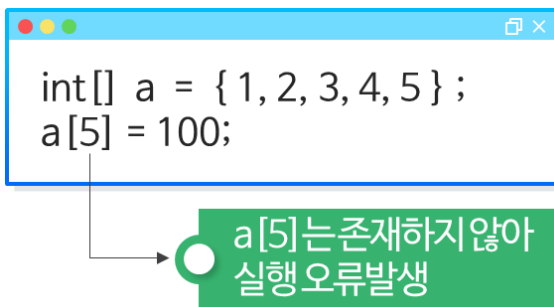
4) 배열 요소의 개수

(1) 배열이름 .length

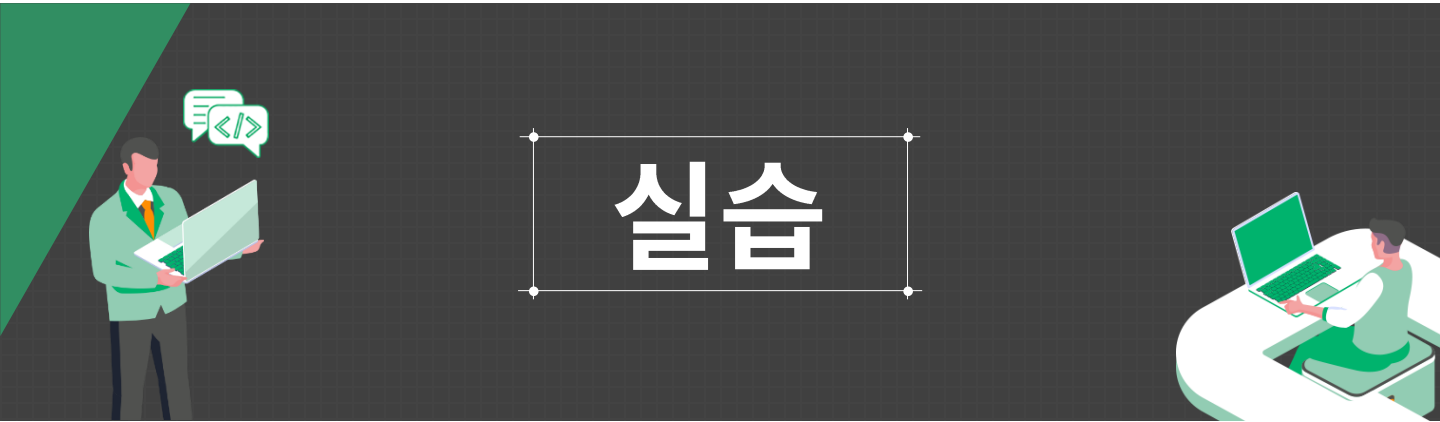
✓ 배열의 멤버 변수(필드)로 배열 요소의 개수를 가지고 있음



(2) 배열의 크기를 초과하여 사용하지 않기



## ⑤ 배열의 기본 문법 이해하기



## 배열 선언과 생성 실습



## 실행 화면

배열 i의 요소 개수는 5  
Exception in thread "main"  
java.lang.ArrayIndexOutOfBoundsException:  
Index 5 out of bounds for length 5  
at  
arrayBasic.ArrayBasic.main (ArrayBasic.java:3  
8)

- 소스 파일명 : [ArrayBasic.java]
- 자세한 내용은 실습 영상을 확인해보세요.

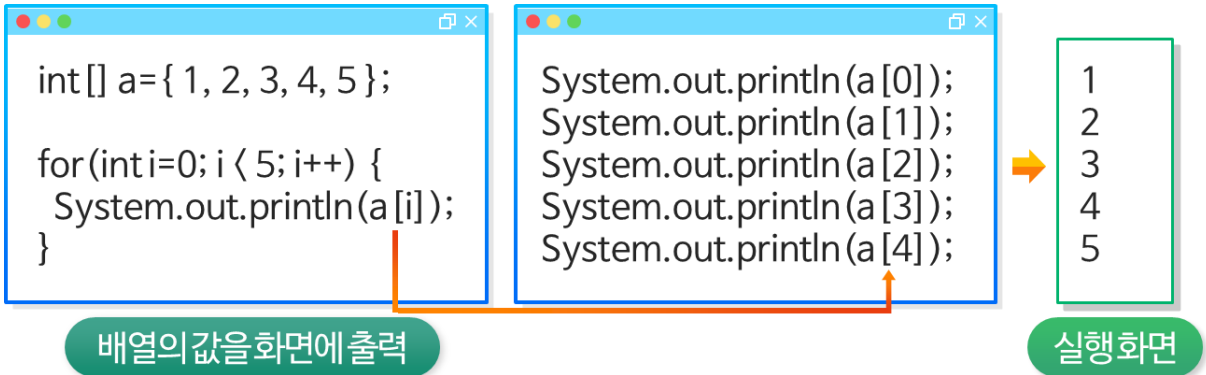
## ⑤ 배열의 기본 문법 이해하기

## 배열 활용하기

## ▶ 배열과 반복문 사용하기

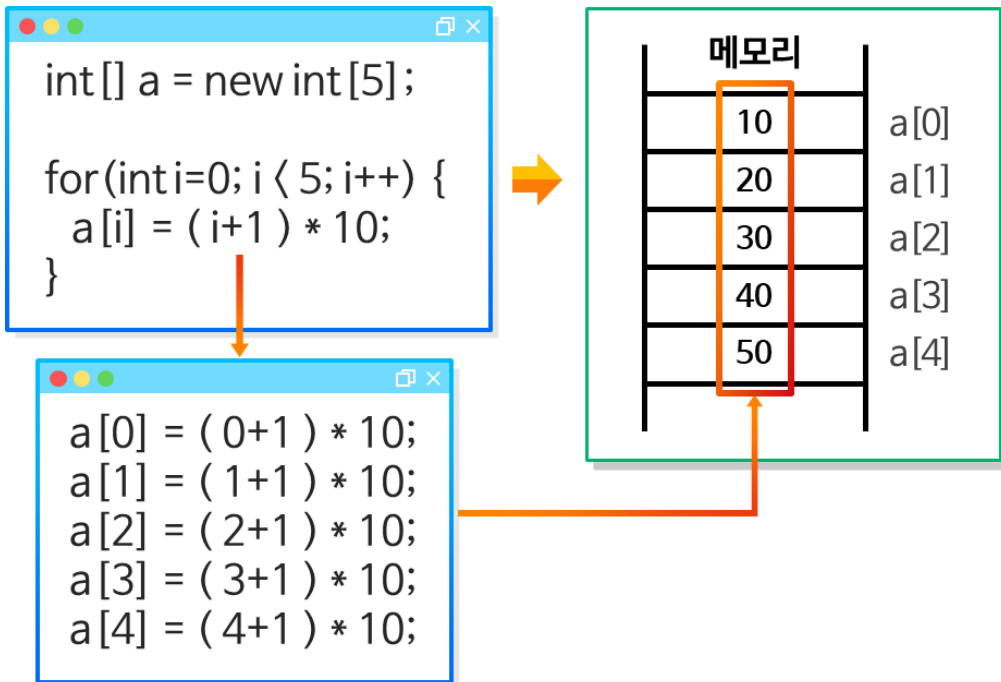
## 1) 배열과 반복문의 관계

- (1) 배열의 인덱스값은 0부터 순차적으로 증가함
- (2) 반복문을 이용하면 인덱스값을 순차적으로 증가할 수 있음



## 2) 배열과 반복문의 예

## (1) 배열의 요소값 저장하기





## ⑤ 배열의 기본 문법 이해하기

## 배열 활용하기

## (2) 배열의 합계와 평균 구하기

```
sum = 0 + a[0];
sum = 34 + a[1];
sum = 44 + a[2];
sum = 69 + a[3];
sum = 140 + a[4];
```

```
int[] a = { 34, 10, 25, 71, 80 };
int sum = 0;

for(int i=0; i < 5; i++) {
    sum = sum + a[i];
}
System.out.println("합계="+sum+"평균="+sum / 5);
```

합계 = 220, 평균 = 44

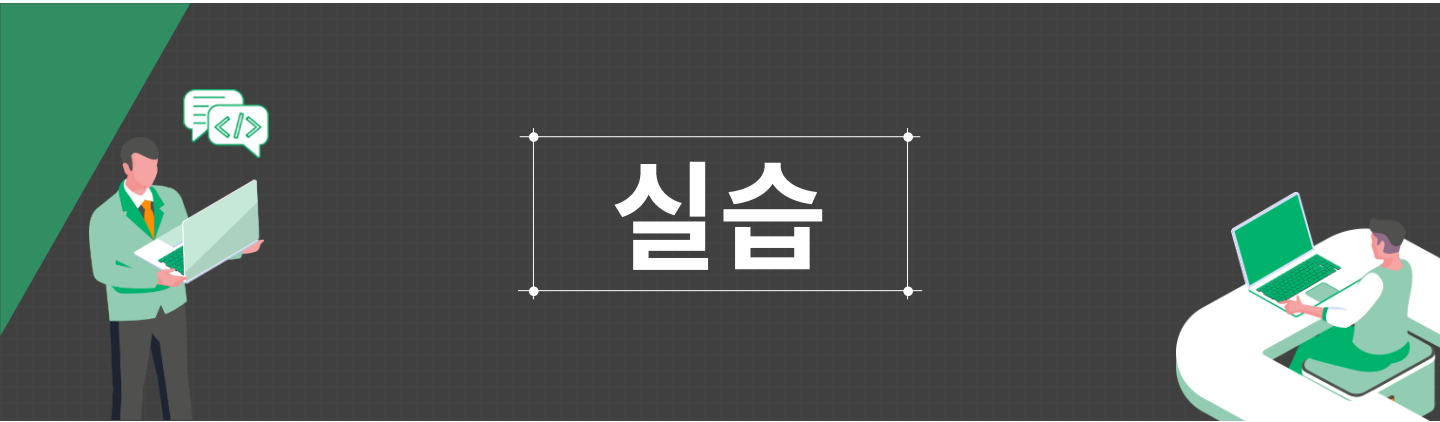
## 3) 배열이름 .length 활용하기

```
int[] a={ 1, 2, 3, 4, 5 };

for(int i=0; i < a.length; i++) {
    System.out.println(a[i]);
}
```

배열요소의개수

## ⑤ 배열의 기본 문법 이해하기



## 배열과 반복문 실습



## 실행 화면

[배열의 값 출력하기]

a[0] = 1

a[1] = 2

a[2] = 3

a[3] = 4

a[4] = 5

[배열의 값 저장후 출력하기]

a[0] = 10

a[1] = 20

a[2] = 30

a[3] = 40

a[4] = 50

[배열의 합계와 평균 구하기] ...

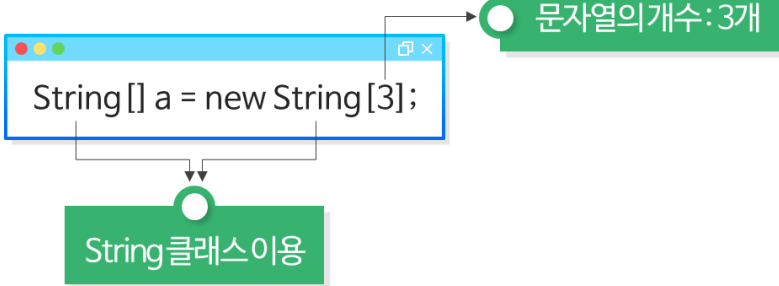
- 소스 파일명 : [ArrayFor.java]
- 자세한 내용은 실습 영상을 확인해보세요.

## ⑤ 배열의 기본 문법 이해하기

## 배열 활용하기

## ▶ 문자열 배열 사용하기

## 1) 문자열 배열 선언 및 생성하기



## 2) 문자열 배열 초기화

```
String [] a = new String [] { "apple", "banana", "melon" };
```

➡ 배열 선언 및 생성과 동시에 배열 요소의 값들을 지정하는 방법

```
String [] a = new String [3];
a[0] = "apple";
a[1] = "banana";
a[2] = "melon";
```

➡ 배열 선언 및 생성 후에  
배열 요소의 값들을 지정하는  
방법

```
String [] a = new String [3];
a[0] = new String ("apple");
a[1] = new String ("banana");
a[2] = new String ("melon");
```

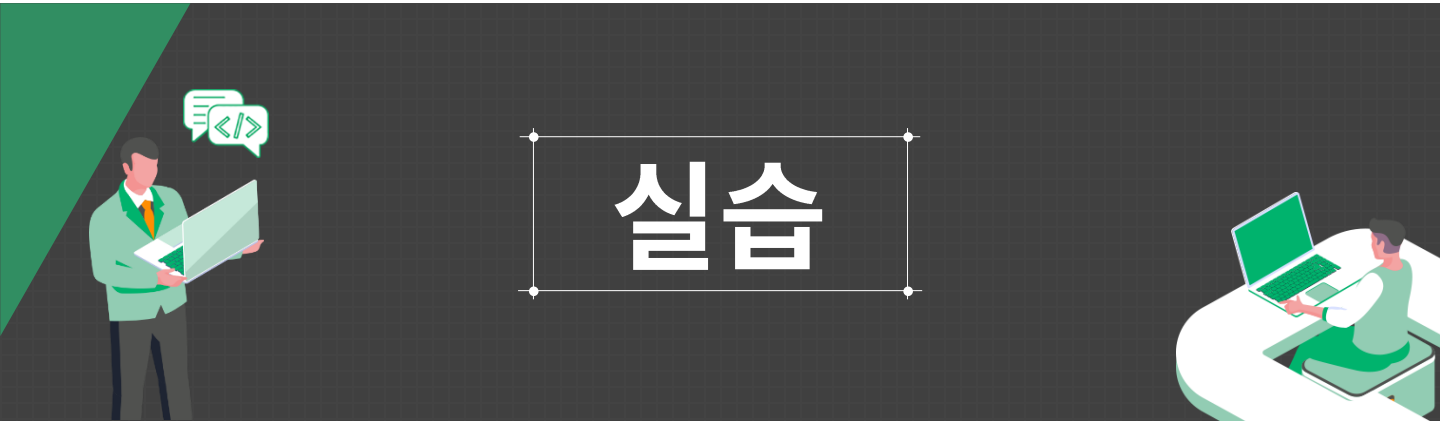
➡ 배열 선언 및 생성 후에  
String 클래스를 이용하여  
메모리를 할당하여 지정

## 3) 문자열 배열 출력하기

```
String [] a = new String [] { "apple", "banana", "melon" };
for(int i=0; i < a.length; i++) {
    System.out.println(a[i]);
}
```

문자열배열 요소별로  
화면에 출력

## ⑤ 배열의 기본 문법 이해하기



## 문자열 배열 실습



## 실행 화면

[과일]

apple banana melon

[책]

수학책 과학책 소설책

[드라마]

sunshine doctor things

1번째 이름을 입력하세요 : 김유신

2번째 이름을 입력하세요 : 이순신

3번째 이름을 입력하세요 : 박열

4번째 이름을 입력하세요 : 조자룡

5번째 이름을 입력하세요 : 유비

[입력한 이름]

김유신 이순신 박열 조자룡 유비

- 소스 파일명 : [ArrayString.java]
- 자세한 내용은 실습 영상을 확인해보세요.

## ⑤ 배열의 기본 문법 이해하기

## 배열 활용하기

## ➤ main() 메소드 인수 처리

1) 실행 시 main() 메소드의 인수로 값이 넘어옴

```

public static void main (String[] args)
{
    for(int i=0; i < args.length; i++) {
        System.out.println(args[i]);
    }
}

```

실행시 문자열배열  
args로 값이넘어옴문자열배열args  
화면에출력

2) 인수 지정하는 방법

(1) [명령 프롬프트] 에서 실행 시 지정하는 방법

명령 프롬프트에서실행

C:\&gt; java ArrayString Hello Java

```

Class ArrayString {
    public static void main (String[] args) {
        System.out.println(args[0]);
        System.out.println(args[1]);
    }
}

```

"Hello"

"Java"

## ⑤ 배열의 기본 문법 이해하기

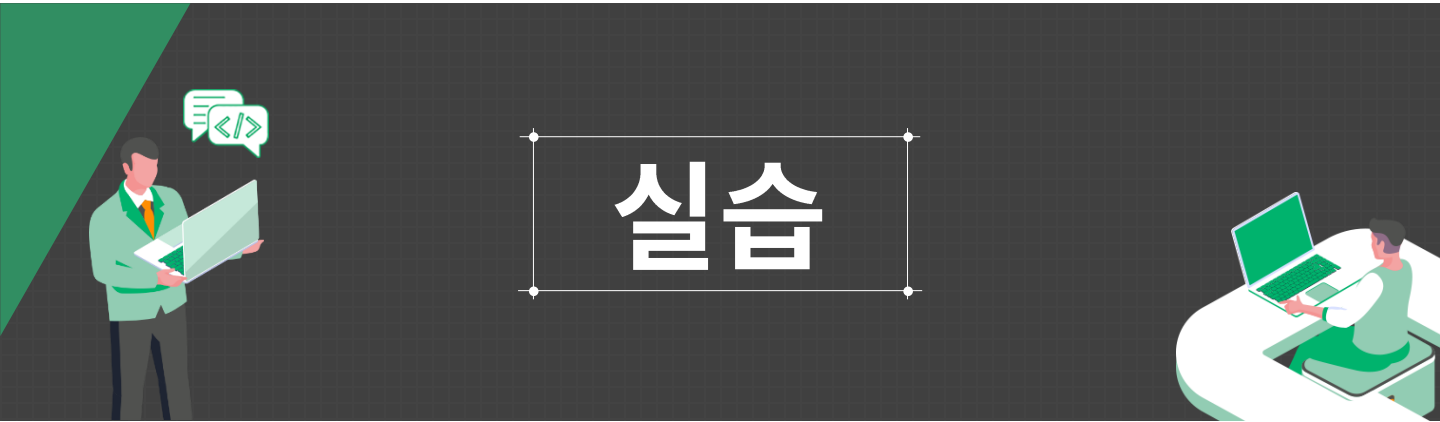
## 배열 활용하기

(2) [이클립스]에서 지정하는 방법

✓ [Run] - [Run configurations] 실행



## ⑤ 배열의 기본 문법 이해하기



## main() 메소드 실습



## 실행 화면

```
D:\WarrayMain\Wbin>java ArrayMain Hello Java  
main() 메소드 인수 값  
1번째 인수의 값은 [Hello] 입니다.  
2번째 인수의 값은 [Java] 입니다.
```

```
D:\WarrayMain\Wbin>
```

- 소스 파일명 : [ArrayMain.java]
- 자세한 내용은 실습 영상을 확인해보세요.

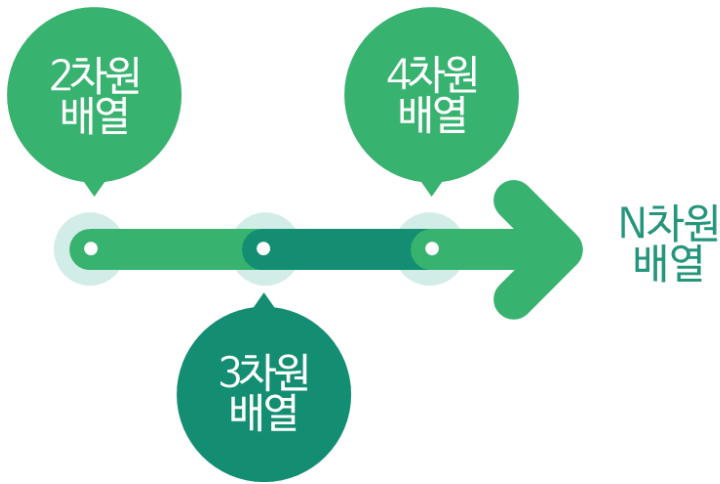
## ⑤ 배열의 기본 문법 이해하기

## 다차원 배열 이해하기

## ▶ 다차원 배열의 개요

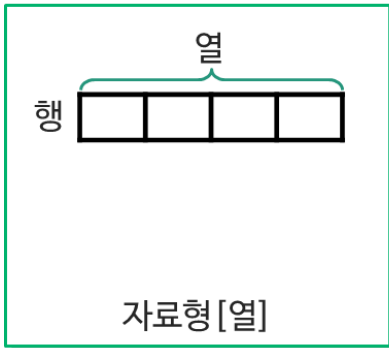
## ▶ 다차원 배열이란?

- (1) 2차원 배열 이상
- (2) 종류

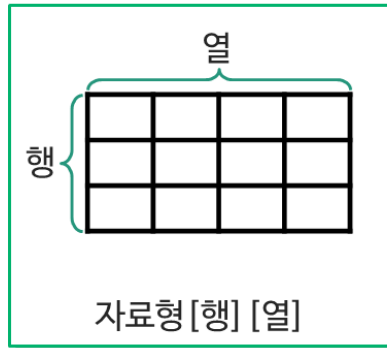


- ✓ 1차원 배열 : 행 1개, 열 N개
- ✓ 2차원 배열 : 행 N개, 열 N개 (1차원 배열이 N개)
- ✓ 3차원 배열 : 층 N개, 행 N개, 열 N개 (2차원 배열이 N개)

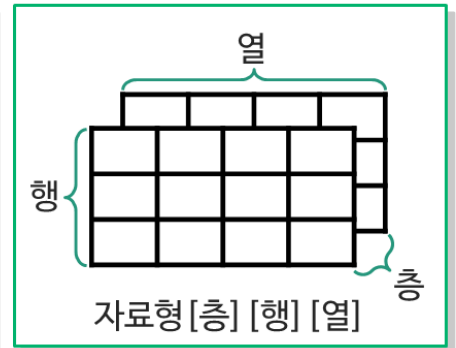
1차원 배열



2차원 배열



3차원 배열





## ⑤ 배열의 기본 문법 이해하기

## 다차원 배열 이해하기

## ➤ 2차원 배열

## 1) 2차원 배열의 선언, 생성, 초기화

## (1) 2차원 배열 선언 및 생성 후 배열 요소에 초기값 지정

```
int[] [] a = new int[3][4];
a[0][0] = 1;
a[0][1] = 2;
⋮
a[2][3] = 12;
```

열(4)

[0][0]	[0][1]	[0][2]	[0][3]
1	2	3	4
[1][0]	[1][1]	[1][2]	[1][3]
5	6	7	8
[2][0]	[2][1]	[2][2]	[2][3]
9	10	11	12

행(3)

## (2) 2차원 배열 선언 및 생성과 동시에 초기값 지정

```
int[] [] a = new int[] [] {
    { 1, 2, 3, 4 },
    { 5, 6, 7, 8 },
    { 9, 10, 11, 12 }
};
```

행(3)

열(4)

○ 각 행의 마지막에 콤마 사용

○ 마지막 행은 콤마 사용 안함

```
int[] [] a = { { 1, 2, 3 }, { 4, 5, 6 } };
```

## ⑤ 배열의 기본 문법 이해하기

## 다차원 배열 이해하기

## 2) 2차원 가변 배열

(1) 행의 개수는 정해져 있고, 열의 개수가 가변임

열						
행(3)	[0][0] 1	[0][1] 2	[0][2] 3	[0][3] 4	[0][4] 5	[0][5] 6
	[1][0] 7	[1][1] 8	[1][2] 9	[1][3] 10		
	[2][0] 11	[2][1] 12	[2][2] 13	[2][3] 14	[2][4] 15	

(2) 선언 및 생성, 초기화

```

int[] [] a = new int[2] [];
a[0] = new int[3];
a[1] = new int[4];
a[0][0] = 1;
a[0][1] = 2;
a[0][2] = 3;
a[1][0] = 4;
a[1][1] = 5;
a[1][2] = 6;
a[1][3] = 7;

```

2차원배열 선언 및  
행의개수지정

열의개수지정

배열 요소별로  
초기값지정

```

int[] [] a = new int[2] [];
a[0] = new int[] { 1,2,3 };
a[1] = new int[] { 4,5,6,7 };

```

```

int[] [] a = new int[][] { {1,2,3}, {4,5,6,7} };

```

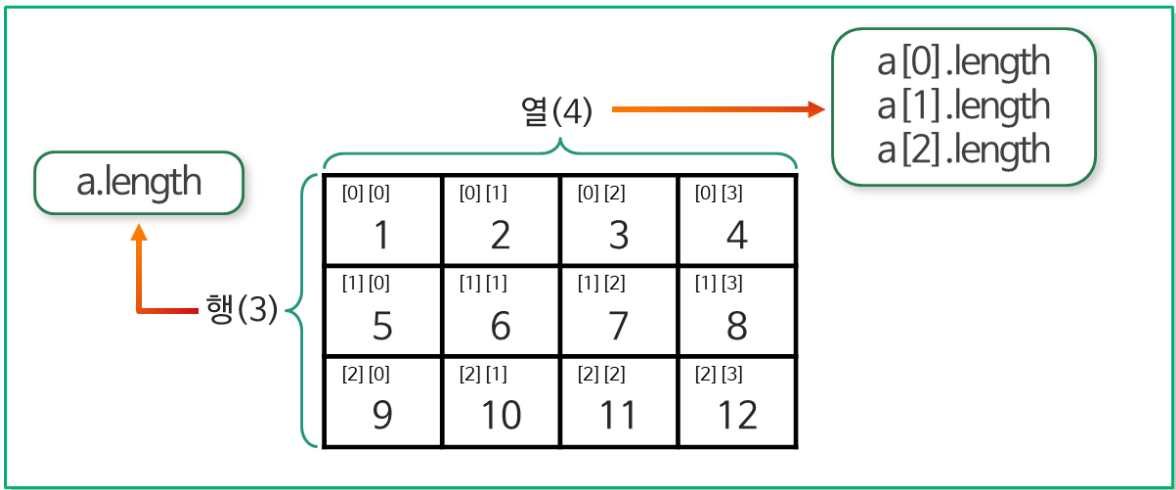
## ⑤ 배열의 기본 문법 이해하기

## 다차원 배열 이해하기

## 3) 2차원 배열의 length 멤버 변수 (필드)

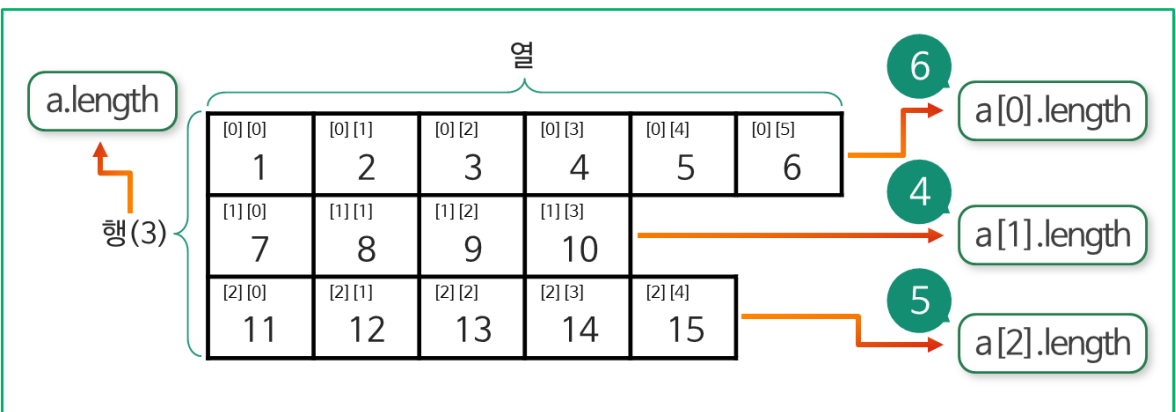
- 배열이름 .length : 2차원 배열의 행의 개수
- 배열이름[행] .length : 2차원 배열의 열의 개수

```
int[] [] a = new int[] [] { {1,2,3,4}, {5,6,7,8}, {9,10,11,12} };
```



- 2차원 가변 배열인 경우

```
int[] [] a = new int[] [] { {1,2,3,4,5,6}, {7,8,9,10}, {11,12,13,14,15} };
```



## ⑤ 배열의 기본 문법 이해하기

## 다차원 배열 이해하기

## 4) 2차원 배열 출력하기

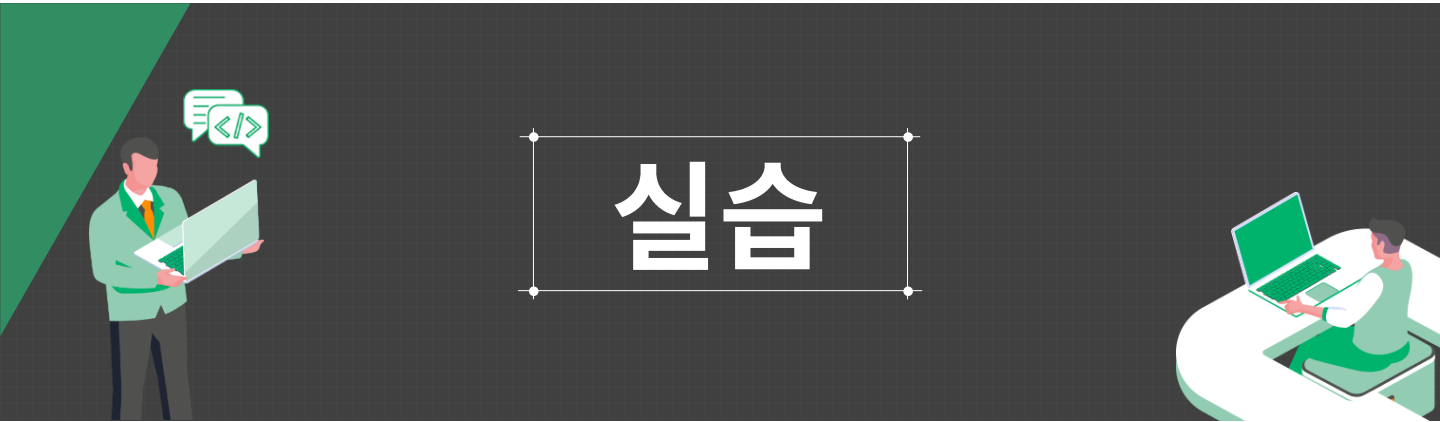
```
int[] [] a = new int[] [] { {1,2,3,4,5,6}, {7,8,9,10}, {11,12,13,14,15} };

for(int row=0; row < a.length; row++) {
    for(int col=0; col < a[row].length; col++) {
        System.out.println(a[row][col]);
    }
}
```

배열 요소별 출력

- 열 반복
- col변수 0부터 각 행의 마지막인덱스까지 반복
- 행 반복
- row변수 0부터 2까지 반복

## ⑤ 배열의 기본 문법 이해하기



### 2차원 배열 실습



#### 실행 화면

```
[2차원 배열 출력]
1 2 3 4
5 6 7 8
9 10 11 12
[2차원 가변 배열 출력]
1 2 3 4 5 6
7 8 9 10
11 12 13 14 15
```

- 소스 파일명 : [ArrayMulti.java]
- 자세한 내용은 실습 영상을 확인해보세요.



## 정리하기

### ■ 배열 이해하기

- 배열이란?
  - 동일한 자료형 여러 개를 묶어서 처리하는 자료구조
  - 변수이름과 인덱스(index)를 이용하여 데이터를 표현
- 배열의 필요성
  - 동일한 자료형 여러 개를 선언하는 경우 변수가 여러 개 필요
  - 배열을 이용하면 하나의 변수만 필요하여 효율적인 프로그래밍 가능
- 배열 선언 방법
  - 배열 선언 시 배열 요소의 개수는 지정하지 않음
  - 형식 : 자료형[] 배열이름; 자료형 배열이름[];
- 배열 생성 방법
  - 배열을 생성하는 것은 메모리를 할당하는 작업임
  - new 연산자를 이용하여 배열에 메모리를 할당하면 배열이 생성됨
  - 배열 생성 시 배열 요소의 개수를 지정함
  - 형식 : 자료형 배열이름[ ] = new 자료형[배열 요소의 개수];
- 배열 초기화 방법
  - 배열 선언 및 생성 후 배열 요소의 값들을 지정하는 방법
  - 배열 선언과 동시에 배열 요소의 값들을 지정하는 방법
  - 배열 선언 후 배열 요소의 값들을 지정하는 방법
- 배열 요소의 개수
  - 배열 이름 .length
    - 배열의 멤버변수로 배열요소의 개수를 가지고 있음



## 정리하기

### ■ 배열 활용하기

- 배열과 반복문 사용하기
  - 배열과 반복문의 관계
    - 배열의 인덱스값은 0부터 순차적으로 증가함
    - 반복문을 이용하면 인덱스값을 순차적으로 증가할 수 있음
  - 배열이름 .length : 배열 요소의 개수를 가짐
- 문자열 배열 사용하기
  - 문자열 배열 선언 및 생성하기: `String[] a = new String[3];`
  - 문자열 배열 초기화: `String[] a = new String[] { "apple", "banana", "melon" };`
- main()메소드 인수 처리
  - 실행시 main() 메소드의 인수로 값이 넘어옴
  - 인수 지정하는 방법
    - [명령 프롬프트]에서 실행 시 지정하는 방법
    - [이클립스]에서 지정하는 방법



## 정리하기

### ■ 다차원 배열 이해하기

- 다차원 배열이란?
  - 2차원 배열 이상
  - 종류
    - 2차원 배열 : 행 N개, 열 N개 (1차원 배열이 N개)
    - 3차원 배열 : 총 N개, 행 N개, 열 N개(2차원 배열이 N개)
  - 2차원 배열 : 2차원 배열 선언 및 생성 후 배열 요소에 초기값 지정
  - 2차원 가변 배열 : 행의 개수는 정해져 있고, 열의 개수가 가변임
  - 2차원 배열의 length 멤버 변수(필드)
    - 배열이름.length : 2차원 배열의 행의 개수
    - 배열이름[행].length : 2차원 배열의 열의 개수