

# *Java* CobinCabin iRaCha 배열

# 1차원 배열

## 배열(array)

- 배열(array)은 같은 타입의 변수가 여러 개 필요한 경우 이용한다.
  - 같은 타입의 변수가 하나의 배열이름으로 연속적인 메모리에 할당된다.
- 배열의 선언
  - 생성된 배열을 다루기 위한 참조변수를 위한 공간 확보
  - 타입[] 변수 이름;
  - 타입 변수이름[];
  - 기능 : 데이터형의 변수를 크기만큼 할당한다.
- 사용 예
  - int[] score; String[] name;
  - int score[]; String name[];

# 1차원 배열

## 배열(array)

- 정수 배열 생성
  - `int [] score;`
  - `score = new int[5];`
  - 각 배열요소는 자동적으로 `int`의 기본값 0으로 초기화 된다
- 1차원 배열의 접근
  - 배열 이름에 인덱스(`[]`)를 이용하여 각각의 변수를 구별
  - 인덱스는 항상 0에서부터 시작
- 배열 객체의 멤버변수 `length`가 배열의 크기에 대한 정보를 담고 있다
  - `score.length`는 5

# 1차원 배열

## 배열(array)

- 참조 변수 배열 생성
  - `String[] name;`
  - `name = new String[3];`
  - 각 배열요소는 자동적으로 null값으로 초기화 된다
- 변수 타입에 따른 기본값

자료형	기본값
Boolean	false
char	'\u0000'
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d 또는 0.0
참조형 변수	null

# 1차원 배열

## 초기화

- 배열은 선언하면서 초기 값을 지정할 수 있다.
- 배열의 초기 값은 여러 개 존재하기 때문에 중괄호({})안에 차례대로 쓴다.
- 초기 값과 배열의 크기가 일치하면 배열의 크기를 생략할 수 있다
- 배열의 크기보다 적은 개수의 초기 값을 할당하게 되면 나머지 배열의 값은 0으로 초기화

CobinCabin  
iRaCha

# 1차원 배열

## 초기화

- int형
  - 0으로 자동 초기화 된다

```
int [] score = new int[5];
```

```
score[0] = 100;
```

```
score[1] = 90;
```

```
score[2] = 80;
```

```
score[3] = 70;
```

```
score[4] = 60
```

```
int [] score = {100, 90, 80, 70, 60};
```

```
int[] score = new int[] {100, 90, 80, 70, 60};
```

# 1차원 배열

## 배열 복사

- 배열은 한 번 생성되면 크기를 변경할 수 없기 때문에 보다 큰 배열을 만들고 이 전 배열로부터 내용을 복사해야 한다.
- 반복문 이용 복사
  - for 문 사용
- System 클래스의 `arraycopy()` 사용
  - 배열에 저장되어 있는 값만 복사
  - 참조 변수 배열인 경우에는 단지 주소 값 만을 복사하고 객체를 복사하지 않는다.

# 다차원 배열

## 2차원 배열

- 2차원 배열은 행과 열로 구성된 형태로, 첫 번째 크기는 행을 의미하며 두 번째 크기는 열을 의미한다.
  - 행과 열은 논리적인 표현일 뿐, 실제 메모리는 1차원 배열과 같은 방식으로 할당
- 2차원 배열의 선언
  - 타입 `[][]` 변수 이름; `int [][] score;`
  - 타입 변수 이름 `[][]`; `int score[][];`
  - 타입 `[]` 변수 이름 `[]`; `int[] score[];`
- 2차원 배열 생성
  - `int [][] score = new int[5][3];`
- 2차원 배열의 접근
  - 행과 열로 구성된 2차원 배열은 인덱스 두 개를 써서 각각의 변수를 구별해야 한다.