

배열

배열의 개념

- 변수에 한가지 값(`literal`)을 대입했다면 이러한 값들을 그룹핑 한 것이 배열이다.
- 변수는 한가지 데이터를 저장, **배열은 여러 개의 데이터를 저장하기 위한 것!**

배열의 생성과 사용

- 아래와 같이 한 반의 학생들 이름을 모두 저장할 수 있다.

```
String[] class1 = {"김윤진", "차동원", "홍길동"};  
System.out.println(class1[0]);  
System.out.println(class1[1]);  
System.out.println(class1[2]);
```

배열의 생성과 사용

- 배열을 생성과 동시에 초기화 할 때는 {}를 사용하며 아래와 같이 우선 생성한 후에 값을 대입 할 수도 있다.

```
String[] class1 = new String[3];  
class1[0] = "김윤진";  
class1[2] = "홍길동";  
class1[1] = "차동원";
```

배열의 생성과 사용

- `DataType[] 변수명` or `DataType 변수명[]` 으로 선언 가능
- 자바에서 배열은 반드시 배열의 **크기가 고정**된다.
- **.length**로 배열의 길이를 얻을 수 있다.

```
String[] class1 = {"김윤진", "홍길동", "차동원"};  
String[] class2 = new String[3];    // class2[0] = "김윤진";  
String[] class3 = new String[] {"김윤진", "홍길동", "차동원"};
```


배열과 반복문

- 배열의 요소를 하나씩 꺼내서 출력하고있다.
- **.length로 배열의 크기만큼 모두 반복**하고 있다.

```
String[] class1 = {"김윤진", "홍길동", "차동원"};

for(int i=0; i<class1.length; i++) {
    System.out.println(class1[i] + "이 상담을 받았습니다.");
}
```

for-each (향상된 for문)

- String[]의 요소 하나 하나는 String이다.
- [index] 로 접근하지 않고 배열의 요소를 바로 사용한다.
- **반복문 내에서 index가 필요 없을 경우** 많이 사용된다.

```
String[] class1 = {"김윤진", "홍길동", "차동원"};

for(String member : class1) {
    System.out.println(member + "이 상담을 받았습니다.");
}
```

배열의 오류와 한계

- 배열의 크기를 넘어서 index에 접근할 경우
- 배열의 크기보다 큰 index에 데이터를 저장하려 할 때

```
String[] class1 = {"김윤진", "홍길동", "차동원"};
```

```
System.out.println( class1[5] );
```

```
class1[3] = "소지섭";
```