



배열

● 배열의 정의

- 자료형이 같은 둘 이상의 값을 저장할 수 있는 메모리 공간
- 배열의 인덱스는 0부터 시작
- 배열에는 1차원 배열과 2차원 이상의 다차원 배열 존재
- 기본 문법

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

- int[] ref : int 형 1차원 배열 인스턴스를 참조할 수 있는 참조변수 ref 선언
- new int[5] : int 형 데이터 5개를 저장할 수 있는 배열 인스턴스 생성
- 생성한 배열 ref는 ref[0], ref[1], ..., ref[5]로 이루어져 있음

배열

```
public class ArrayInsInstance {
    public static void main(String[] args) {
        // 길이가 5인 int 형 1차원 배열 생성
        int[] ar1 = new int[5];
        for (int i = 0; i < 5; i++) {
            ar1[i] = i + 5;
        }
        double[] ar2 = new double[7];
        for (int i = 0; i < 7; i++) {
            ar2[i] = (double) i + 0.5;
        }
        System.out.println("배열 ar1의 길이 : " + ar1.length);
        System.out.println("배열 ar2의 길이 : " + ar2.length);
        for (int i = 0; i < 5; i++) {
            System.out.println("배열 ar1[" + i + "] : " + ar1[i]);
        }
        for (int i = 0; i < 7; i++) {
            System.out.println("배열 ar2[" + i + "] : " + ar2[i]);
        }
    }
}
```

배열

- 인스턴스를 저장할 수 있는 배열의 생성

```
public class Box {  
    private String conts;  
    public Box(String conts) {  
        this.conts = conts;  
    }  
    public String toString() {  
        return this.conts;  
    }  
}
```

```
public class ArrayInsInstance2 {  
    public static void main(String[] args) {  
        Box[] ar = new Box[5];  
        System.out.println("length : " + ar.length);  
    }  
}
```

배열

- 배열을 대상으로 한 값의 저장과 참조

```
public class BoxArray {  
    public static void main(String[] args) {  
        Box[] ar = new Box[3];  
  
        // 배열에 인스턴스 저장  
        ar[0] = new Box("First");  
        ar[1] = new Box("Second");  
        ar[2] = new Box("Third");  
  
        // 저장된 인스턴스 참조  
        System.out.println(ar[0]);  
        System.out.println(ar[1]);  
        System.out.println(ar[2]);  
    }  
}
```

배열

- 배열을 생성과 동시에 초기화

```
public class InitArray {  
    public static void main(String[] args) {  
        int[] arr = {1, 2, 3};  
        for (int i = 0; i < 3; i++) {  
            System.out.println(arr[i]);  
        }  
    }  
}
```

- 배열 참조변수 선언의 두 가지 방법

- int[] arr
- int arr[]

배열

- 강화된 for문

```
public class EnhancedFor {  
    public static void main(String[] args) {  
        int[] ar = {1, 2, 3, 4, 5};  
  
        // 배열 요소 전체 출력  
        for (int e : ar) {  
            System.out.println(e);  
        }  
  
        // 배열 요소의 전체 합 출력  
        int sum = 0;  
        for (int e : ar) {  
            sum += e;  
        }  
        System.out.println("sum = " + sum);  
    }  
}
```

배열

- 인스턴스 배열을 대상으로 강화된 for문

```
public class Box1 {  
    private String contents;  
    private int boxNum;  
  
    public Box1(String contents, int boxNum) {  
        this.contents = contents;  
        this.boxNum = boxNum;  
    }  
  
    public int getBoxNum() {  
        return this.boxNum;  
    }  
  
    @Override  
    public String toString() {  
        return "Box1{" +  
            "contents='" + contents + 'W' +  
            ", boxNum=" + boxNum +  
            '}';  
    }  
}
```



```
public class EnhancedForInst {  
    public static void main(String[] args) {  
        Box1[] ar = new Box1[5];  
        ar[0] = new Box1(101, "Coffe");  
        ar[1] = new Box1(202, "Computer");  
        ar[2] = new Box1(303, "Apple");  
        ar[3] = new Box1(404, "Dress");  
        ar[4] = new Box1(505, "Fairy-tale book");  
  
        for (Box1 e : ar) {  
            if (e.getBoxNum() == 505) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```