



배열

안 화 수

배열

❖ 배열(Array)

동일한 자료형의 데이터를 저장하기 위한 정적인 자료구조

❖ 배열의 특징

1. 배열은 같은 자료형의 데이터만 저장할 수 있다.
2. 한 번 생성된 배열은 길이를 늘리거나 줄일 수 없다.

배열

❖ 1차원 배열 생성

■ 방법1. new 연산자로 배열 생성

배열에 저장될 값이 정해져 있지 않는 경우에 주로 사용하는 형식이다.

```
int[]    score    =    new    int[3];
```

자료형	배열변수	연산자	자료형[배열의 크기]
-----	------	-----	-------------

```
score[0]=80;  
score[1]=90;  
score[2]=100;
```

배열

❖ 1차원 배열 생성

- new 연산자로 배열을 생성하면 heap메모리 영역에 새로운 공간을 할당 받으면서, 배열의 원소들은 자동으로 초기화 된다.

분류	데이터 타입	초기값
기본 타입 (정수)	byte[]	0
	char[]	'\u0000'
	short[]	0
	int[]	0
	long[]	0L
기본 타입 (실수)	float[]	0.0F
	double[]	0.0
기본 타입 (논리)	boolean[]	false
참조 타입	클래스[]	null
	인터페이스[]	null

배열

❖ 1차원 배열 생성

■ 방법2. 값 목록으로 배열 생성

배열에 저장될 값이 정해져 있는 경우에 주로 사용하는 형식이다.
배열 선언과 동시에 초기화를 수행한다.

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

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

```
double[] d = { 3.14, 10.5, 42.195 };
```

```
char[] c = { 'j', 'a', 'v', 'a', '안' };
```

```
String[] str = { "java", "jsp", "oracle" };
```

```
String[] str = new String[]{ "java", "jsp", "oracle" };
```

배열

❖ 배열 예제1 (1/4)

```
public class ArrayEx01 {
```

```
    public static void main(String[] args) {
```

// 1차원 배열 첫번째 형식 : 배열에 저장될 값이 정해져 있지 않는 경우에 주로 사용하는 형식

```
//      int[]   score   =   new      int[3];  
//      자료형   배열변수       연산자   자료형[배열의 크기]
```

// int형 배열은 자동으로 0으로 초기화가 된다.

```
System.out.println(score[0]);    // 0  
System.out.println(score[1]);    // 0  
System.out.println(score[2]);    // 0
```

```
score[0] = 80;  
score[1] = 90;  
score[2] = 100;  
System.out.println(score[0]);    // 80  
System.out.println(score[1]);    // 90  
System.out.println(score[2]);    // 100
```

// double 배열은 자동으로 0.0으로 초기화가 된다.

```
double[] d = new double[3];  
System.out.println(d[0]);        // 0.0  
System.out.println(d[1]);        // 0.0  
System.out.println(d[2]);        // 0.0
```

❖ 배열 예제1 (2/4)

// char 배열은 자동으로 초기화가 되지 않는다.

```
char[] c = new char[3];  
System.out.println(c[0]);  
System.out.println(c[1]);  
System.out.println(c[2]);
```

```
boolean[] b = new boolean[3];  
System.out.println(b[0]);           // false  
System.out.println(b[1]);           // false  
System.out.println(b[2]);           // false
```

```
String[] str = new String[3];  
System.out.println(str[0]);          // null  
System.out.println(str[1]);          // null  
System.out.println(str[2]);          // null
```

```
str[0] = "자바";  
str[1] = "오라클";  
str[2] = "스프링";  
System.out.println(str[0]);          // 자바  
System.out.println(str[1]);          // 오라클  
System.out.println(str[2]);          // 스프링
```

배열

❖ 배열 예제1 (3/4)

// 1차원 배열 두번째 형식 : 배열 선언과 동시에 초기화를 할때 주로 사용되는 형식
// (배열에 할당될 값이 정해져 있는 경우에 주로 사용함)

```
int[] s = {80, 90, 100};  
int[] s1 = new int[] {80, 90, 100};  
System.out.println(s[0]);           // 80  
System.out.println(s[1]);           // 90  
System.out.println(s[2]);           // 100  
  
System.out.println("배열의 크기:" + s.length); // 배열의 크기:3  
for(int i=0; i<s.length; i++)  
    System.out.print(s[i]+"\\t");  
System.out.println();  
  
double[] dd = {3.14, 10.5, 42.195, 50};  
for(int i=0; i<dd.length; i++)  
    System.out.print(dd[i]+"\\t");  
System.out.println();
```


❖ 배열 예제1 (4/4)

```
char[] cc = {'j', 'a', 'v', 'a', '자', '바'};  
for(int i=0; i<cc.length; i++)  
    System.out.print(cc[i]+"\\t");  
System.out.println();
```

```
boolean[] bb = {true, false, true};  
for(int i=0; i<bb.length; i++)  
    System.out.print(bb[i]+"\\t");  
System.out.println();
```

```
String[] str1 = {"자바", "오라클", "스프링", "파이썬", "텐스플로우"};  
String[] str2 = new String[]{"자바", "오라클", "스프링", "파이썬", "텐스플로우"};  
for(int i=0; i<str1.length; i++)  
    System.out.print(str1[i]+"\\t");
```

```
}
```

```
}
```

❖ 배열 예제2

```
public class ArrayEx02 {  
  
    public static void main(String[] args) {  
  
        int[] score = {83, 90, 87};  
        System.out.println("score[0]:"+score[0]);  
        System.out.println("score[1]:"+score[1]);  
        System.out.println("score[2]:"+score[2]);  
  
        int sum = 0;  
        for(int i=0; i<score.length; i++)  
            sum += score[i];                // sum = sum + score[i]  
  
        System.out.println("총점:"+sum); // 총점:260  
  
        double avg = (double)sum/3.0;    // 강제 형변환  
        System.out.println("평균:"+avg); // 평균:86.66666666666667  
  
        // 평균값을 소숫점 2째자리까지 출력 하세요?  
        System.out.printf("평균: %.2f", avg); // 평균: 86.67  
    }  
}
```

❖ 배열 예제3

```
public class ArrayEx03 {  
  
    public static void main(String[] args) {  
  
        // 키보드로 5과목의 점수를 입력 받아서, 총점과 평균을 구하는 프로그램을 작성 하세요?  
  
        int[] s = new int[5];  
  
        System.out.println("5과목의 점수를 입력하세요?");  
        Scanner sc = new Scanner(System.in);  
  
        int sum = 0;  
        for(int i=0; i<s.length; i++) {  
            s[i] = sc.nextInt();  
            sum += s[i];           // sum = sum + s[i]  
        }  
        double avg = sum /5.0;  
        System.out.println("총점:" + sum);  
        System.out.println("평균:" + avg);  
    }  
}
```

❖ 배열 예제4

```
public class ArrayEx04 {  
  
    public static void main(String[] args) {  
  
        // 배열에 저장된 데이터 중에서 최대값과 최소값을 구하는 프로그램을 작성하세요?  
  
        double[] data = {9.5, 7.0, 13.6, 7.5, 10.5};  
  
        double max, min;  
  
        max = data[0];        // 9.5  
        min = data[0];        // 9.5  
  
        for(int i=1; i<data.length; i++) {  
            if(data[i] > max)    max = data[i];  
            if(data[i] < min)    min = data[i];  
        }  
        System.out.println("max:" + max);  
        System.out.println("min:" + min);  
    }  
}
```

❖ 배열 예제5

```
public class ArrayEx05 {  
  
    public static void main(String[] args) {  
        int[] scores;  
        scores = new int[] {83, 90, 87};  
  
        int sum1 = 0;  
        for(int i=0; i<3; i++) {  
            sum1 += scores[i];  
        }  
        System.out.println("총합:" + sum1);           // 총합: 260  
  
        // add메소드를 호출해서 리턴된 총점을 sum2에 저장  
        int sum2 = add(new int[] {83, 90, 87});       // add메소드 호출  
        System.out.println("총합:" + sum2);           // 총합: 260  
  
    } // main() end  
  
    // 사용자 정의 메소드 : 합을 구해주는 역할  
    public static int add(int[] scores) {  
        int sum = 0;  
        for(int i=0; i<3; i++) {  
            sum += scores[i];  
        }  
        return sum;  
    }  
}
```

❖ 향상된 for문

```
for ( 변수 : 순차적인 자료구조(배열, 컬렉션) ) {  
    반복 실행할 문장;  
}
```

배열

❖ 배열 예제6

```
public class ArrayEx06 {  
  
    public static void main(String[] args) {  
  
        // 향상된 for문, 확장 for문  
        // 형식 : for( 변수 : 순차적인 자료구조(배열, 컬렉션) ){  
        //             실행될 문장;  
        //             }  
  
        int[] score = {95, 71, 84, 93, 87};  
  
        //      1.기본 for문  
        int sum = 0;  
        for(int i=0; i<score.length; i++)  
            sum += score[i];  
        System.out.println("총합:" + sum);  
  
        //      2.향상된 for문  
        int sum1 = 0;  
        for(int s : score)  
            sum1 += s;  
        System.out.println("총합:" + sum1);  
    }  
}
```

❖ 배열 예제7

```
public class ArrayEx07 {  
  
    // 절대값을 구해주는 메소드  
    static int abs(int data) {  
        if(data < 0)  
            data = -data;  
  
        return data;  
    }  
  
    public static void main(String[] args) {  
  
        // args[0] = "-10", args[1] = "-20"  
        System.out.println("args[0]:" + args[0]);  
        System.out.println("args[1]:" + args[1]);  
  
        // 문자를 숫자로 형변환 : "-10" --> -10  
        int num = Integer.parseInt(args[0]);  
        System.out.println("절대값:" + abs(num)); // abs() 메소드 호출  
    }  
}
```


배열

❖ 배열 예제8

```
public class ArrayEx08 {  
  
    public static void main(String[] args) {  
  
        // 배열 복사 : for문으로 배열 복사  
  
        int[] oldArray = {10, 20, 30};           // 원본 배열  
        int[] newArray = new int[5];           // 새로운 배열  
  
        for(int i=0; i<oldArray.length; i++) {  
            newArray[i] = oldArray[i];           // 배열 복사  
        }  
  
        // 복사된 배열 출력  
        for(int i : newArray) {  
            System.out.print(i+"\t");  
        }  
    }  
}
```

2차원 배열

❖ 2차원 배열 생성

■ 방법1. new 연산자로 배열 생성

배열에 저장될 값이 정해져 있지 않는 경우에 주로 사용하는 형식이다.

```
int[][] score = new int[5][3];
```

자료형	배열변수	연산자	자료형	[5]:행
				[3]:열

-> 정수값을 저장할 수 있는 5행 3열짜리 기억 공간을 생성

```
score[0][0]=85;  score[0][1]=60;  score[0][2]=70;
score[1][0]=90;  score[1][1]=95;  score[1][2]=80;
score[2][0]=75;  score[2][1]=80;  score[2][2]=100;
score[3][0]=80;  score[3][1]=70;  score[3][2]=95;
score[4][0]=100; score[4][1]=65;  score[4][2]=80;
```

2차원 배열

❖ 배열 예제

```
public class Arr03 {  
  
    public static void main(String[] args)  
    {  
        //정수값을 저장할 수 있는 5행 3열짜리 기억 공간 생성  
        int [][]score=new int [5][3];  
  
        //행과 열의 위치에 값 할당  
        score[0][0]=85;   score[0][1]=60;   score[0][2]=70;  
        score[1][0]=90;   score[1][1]=95;   score[1][2]=80;  
        score[2][0]=75;   score[2][1]=80;   score[2][2]=100;  
        score[3][0]=80;   score[3][1]=70;   score[3][2]=95;  
        score[4][0]=100;  score[4][1]=65;   score[4][2]=80;  
  
        for(int row = 0; row < 5; row++){           // 학생수 (1~5번:5명)  
            for(int col = 0; col < 3; col++){         // 과목수 (국어, 영어, 수학)  
                System.out.print(" " + score[row][col]);  
            }  
            System.out.println("");  
        }  
    }  
}
```

2차원 배열

❖ 2차원 배열 생성

■ 방법2. 값 목록으로 배열 생성

배열에 저장될 값이 정해져 있는 경우에 주로 사용하는 형식이다.
배열 선언과 동시에 초기화를 수행한다.

```
int[][] score = { { 85, 60, 70},      // 0 행  
                  { 90, 95, 80},      // 1 행  
                  { 75, 80, 100},     // 2 행  
                  { 80, 70, 95},      // 3 행  
                  {100, 65, 80} };    // 4 행
```

2차원 배열

❖ 배열 예제 (1/2)

```
public class Arr04 {  
  
    public static void main(String[] args) {  
  
        int [][]score = { { 85, 60, 70}, //0 행  
                           { 90, 95, 80}, //1 행  
                           { 75, 80, 100}, //2 행  
                           { 80, 70, 95}, //3 행  
                           {100, 65, 80}  //4 행  
                        };  
  
        int [] subject = new int[3];    //과목총점 저장  
        int [] student = new int[5];    //학생의 총점 저장  
        // subject[0]=0, subject[1]=0, subject[2]=0
```

2차원 배열

❖ 배열 예제 (2/2)

```
System.out.println("각 과목별 총점구하기 ");
for(int c = 0; c < 3 ; c++){          // 과목
    for(int r = 0; r < 5 ; r++){      // 학생
        subject[c] += score[r][c];
    } //subject[c]=subject[c]+score[r][c];
    System.out.println(subject[c]);
}
```

```
System.out.println("학생별 총점구하기");
for(int r = 0; r < 5 ; r++){          // 학생
    for(int c = 0; c < 3 ; c++){      // 과목
        student[r] += score[r][c];
    } //student[r]=student[r]+score[r][c];
    System.out.println(student[r]);
}
```

```
}
```

```
}
```