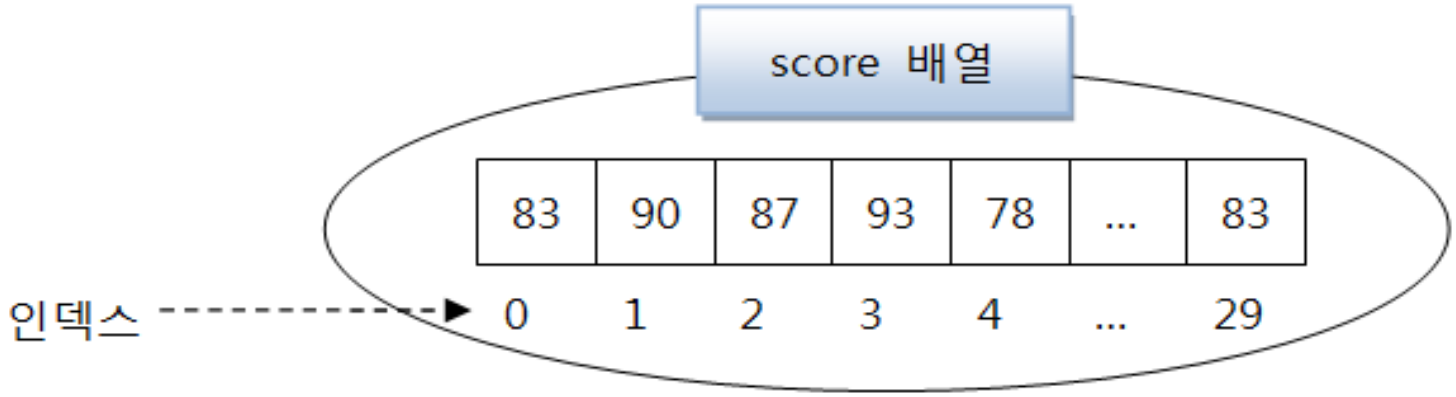
배열 타입

배열 타입

❖ 배열이란?

- 같은 타입의 데이터를 연속된 공간에 저장하는 자료구조
- 각 데이터 저장 위치는 인덱스 부여해 접근

```
int score1= 83;
int score2 = 90;
int score3 = 87;
 :
int score30= 75;
```



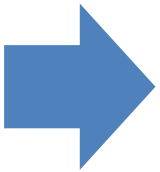
항목 접근: 배열이름[인덱스]
ex) score[0], score[3]

배열 타입

❖ 배열의 장점

- 중복된 변수 선언 줄이기 위해 사용
- 반복문 이용해 요소들을 쉽게 처리

```
int sum = score1;
sum += score2;
sum += score3;
   :
sum += score30;
int avg = sum / 30;
```



```
int sum = 0;
for(int i=0; i<30; i++) {
   sum += score[i];
}
int avg = sum / 30;
```

배열 타입

❖ 배열 선언

- 배열을 사용하기 위해 우선 배열 변수 선언

```
타입[ ] 변수;
```

```
int[] intArray;
double[] doubleArray;
String[] strArray;
```

```
타입 변수[ ];
```

```
int intArray[];
double doubleArray[];
String strArray[];
```

- 배열 변수는 참조 변수 - 배열 생성되기 전 null로 초기화 가능

```
타입[] 변수 = null;
```

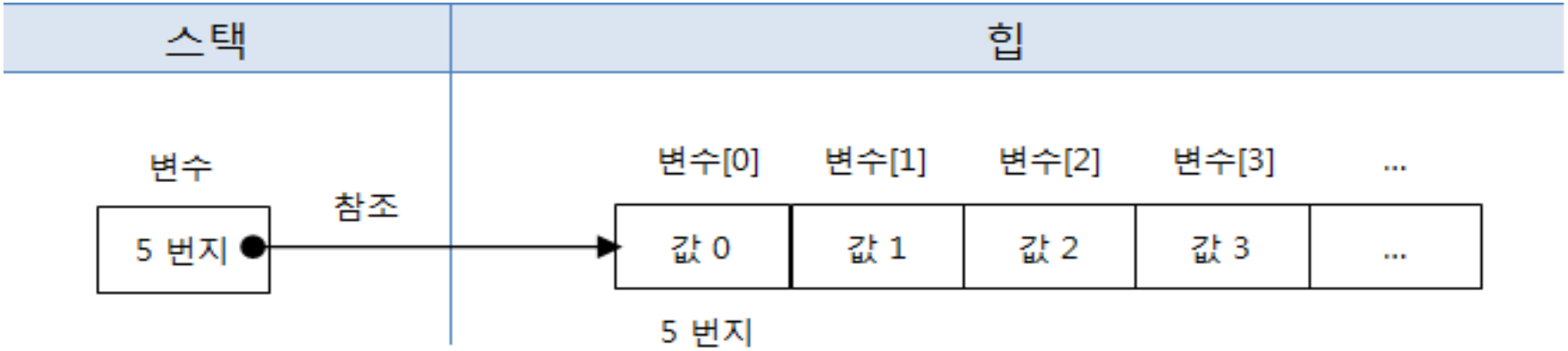
  - 배열 변수가 null 값을 가진 상태에서 항목에 접근 불가
    - 변수[인덱스] 읽기 못함
    - NullPointerException 발생

배열 타입

❖ 값 목록으로 배열 생성하는 방법

- 변수 선언과 동시에 값 목록 대입

```
데이터타입[] 변수 = {값0, 값1, 값2, 값3, … };
```



- 변수 선언 후 값 목록 대입
  - 배열 객체 생성 구문(new ~) 필요

```
데이터타입[] 변수;
변수 = new 타입[] {값0, 값1, 값2, 값3, … };
```

❖ 값 목록으로 배열 생성 : ArrayCreateByValueListExample1.java

```
public class ArrayCreateByValueListExample1 {
   public static void main(String[] args) {
      int[] scores = { 83, 90, 87 };

      System.out.println("scores[0] : " + scores[0]);
      System.out.println("scores[1] : " + scores[1]);
      System.out.println("scores[2] : " + scores[2]);

      int sum = 0;
      for (int i = 0; i < 3; i++) {
         sum += scores[i];
      }
      System.out.println("총합 : " + sum);
      double avg = (double) sum / 3;
      System.out.println("평균 : " + avg);
   }
}
```

❖ 값의 리스트로 배열 생성 : ArrayCreateByValueListExample2.java

```
public class ArrayCreateByValueListExample2 {
   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);

      int sum2 = add(new int[] { 83, 90, 87 });

      System.out.println("총합 : " + sum2);
      System.out.println();
   }
```

❖ 값의 리스트로 배열 생성 : ArrayCreateByValueListExample2.java

```
   public static int add(int[] scores) {
      int sum = 0;
      for (int i = 0; i < 3; i++) {
         sum += scores[i];
      }
      return sum;
   }
}
```

배열 타입

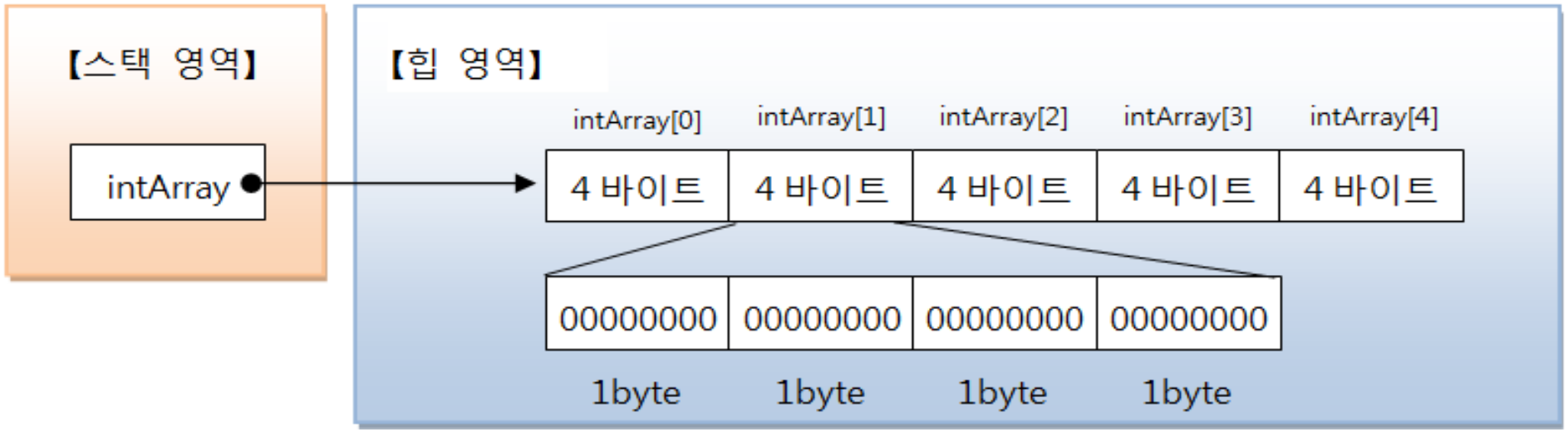
❖ new 연산자로 배열 생성

- ㅇ 배열 생성시 값 목록을 가지고 있지 않음
- ㅇ 향후 값들을 저장할 배열을 미리 생성하고 싶을 경우

```
타입[] 변수 = new 타입[길이];

타입[] 변수 = null;
변수 = new 타입[길이];
```

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



배열 타입

❖ 타입 별 항목의 기본값

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

❖ new 연산자로 배열 생성: ArrayCreateByNewExample.java

```
public class ArrayCreateByNewExample {

   public static void main(String[] args) {

      int[] arr1 = new int[3];

      for (int i = 0; i < 3; i++) {
         System.out.println("arr1[" + i + "] : " + arr1[i]);
      }

      arr1[0] = 10;
      arr1[1] = 20;
      arr1[2] = 30;
      for (int i = 0; i < 3; i++) {
         System.out.println("arr1[" + i + "] : " + arr1[i]);
      }
```

❖ new 연산자로 배열 생성: ArrayCreateByNewExample.java

```
        double[] arr2 = new double[3];
        for (int i = 0; i < 3; i++) {
           System.out.println("arr2[" + i + "] : " + arr2[i]);
        }
        arr2[0] = 0.1;
        arr2[1] = 0.2;
        arr2[2] = 0.3;
        for (int i = 0; i < 3; i++) {
           System.out.println("arr2[" + i + "] : " + arr2[i]);
        }

        String[] arr3 = new String[3];
        for (int i = 0; i < 3; i++) {
           System.out.println("arr3[" + i + "] : " + arr3[i]);
        }
        arr3[0] = "1월";
        arr3[1] = "2월";
        arr3[2] = "3월";
        for (int i = 0; i < 3; i++) {
           System.out.println("arr3[" + i + "] : " + arr3[i]);
        }
    }
}
```

배열 타입

❖ 배열의 길이

- 배열에 저장할 수 있는 전체 항목 수
- 코드에서 배열의 길이 얻는 방법

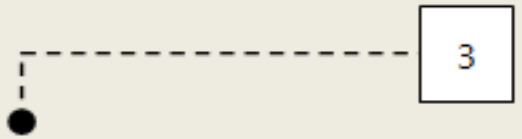
배열변수.length

```
int[] intArray = { 10, 20, 30};
int len = intArray.length;   // 3
```

- 배열의 길이는 읽기 전용
- 배열의 길이는 for문의 조건식에서 주로 사용

```
int[] scores = { 83, 90, 87 };

int sum = 0;
for(int i=0; i<scores.length; i++) {
   sum += scores[i];
}
System.out.println("총합 : " + sum);
```



❖ 배열의 length 필드: ArrayLengthExample.java

```
public class ArrayLengthExample {
   public static void main(String[] args) {
      int[] scores = { 83, 90, 87 };

      int sum = 0;
      for (int i = 0; i < scores.length; i++) {
         sum += scores[i];
      }
      System.out.println("총합 : " + sum);

      double avg = (double) sum / scores.length;
      System.out.println("평균 : " + avg);
   }
}
```

배열 타입

❖ 커맨드 라인 입력

- 배열의 선언과 사용

```
java 클래스 문자열 0  문자열 1  문자열 2  ...  문자열 n-1

String[] args = { 문자열 0, 문자열 1, ... , 문자열 n-1 };

    man() 메소스 호출시 전달

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

❖ 커맨드 라인 입력 : MainStringArrayArgument.java

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

      if (args.length != 2) {   // 입력된 데이터 개수 검사
         System.out.println("프로그램의 사용법");
         System.out.println("java MainStringArrayArgument num1 num2");
         System.exit(0);     // 프로그래미 강제 종료
      }

      String strNum1 = args[0]; // 첫번째 데이터 얻기
      String strNum2 = args[1]; // 두번째 데이터 얻기

      int num1 = Integer.parseInt(strNum1);// 문자열을 정수로 변환
      int num2 = Integer.parseInt(strNum2);// 문자열을 정수로 변환

      int result = num1 + num2;
      System.out.println(num1 + " + " + num2 + " = " + result);
   }
}
```