



# Java 1

4장 배열



# Contents

01. 배열

02. 배열과 반복문

03. 이차원 배열

01

# 배열



# 01 배열

## ✓ 그냥 성적처리

```
int korElice
```

(Elice의 국어성적)

30

```
int korJane
```

(Jane의 국어성적)

85

```
int korBob
```

(Bob의 국어성적)

90

```
int korHarry
```

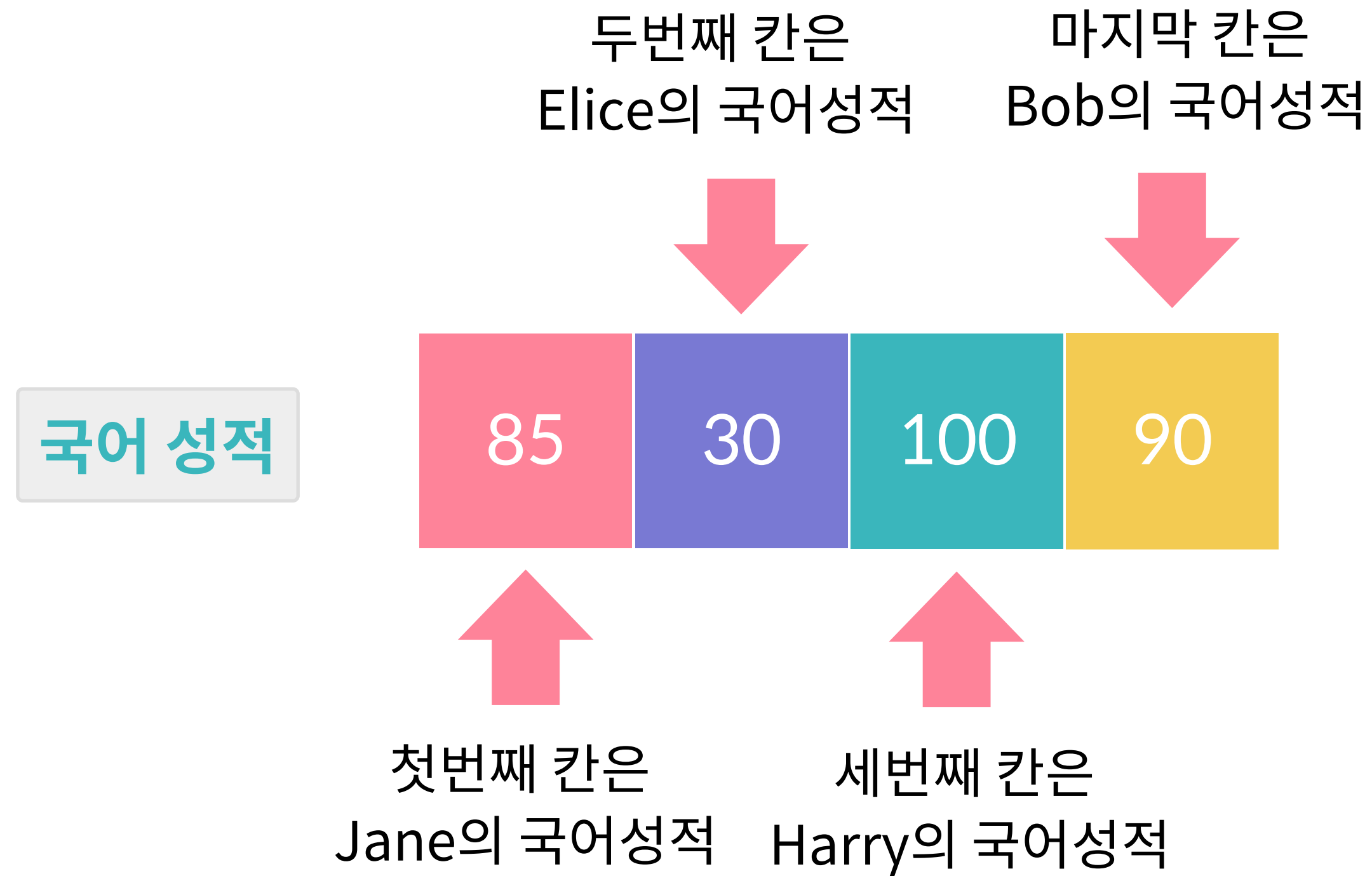
(Harry의 국어성적)

100

`/* elice */`

# 01 배열

## ✔ 조금 바꾼 성적처리!



`/* elice */`

# 01 배열

## ✓ 배열이란?

동일한 자료형의 데이터 여러 개를  
저장하는 자료구조  
배열의 이름은 한 개

int

1	2	3	5	8	6	0	4	7	9
---	---	---	---	---	---	---	---	---	---

/\* elice \*/

# 01 배열

## ✓ 배열이란?

동일한 자료형의 데이터 여러 개를  
저장하는 자료구조  
배열의 이름은 한 개

String

“spring”	“summer”	“fall”	“winter”
----------	----------	--------	----------

# 01 배열

## ✓ 배열이란?

동일한 자료형의 데이터 여러 개를  
저장하는 자료구조  
배열의 이름은 한 개

double

95.3	-15.7	0.0	30.1
------	-------	-----	------



# 01 배열

## ✓ 배열이란?

동일한 자료형의 데이터 여러 개를  
저장하는 자료구조  
배열의 이름은 한 개

boolean

true	true	false	true
------	------	-------	------

# 01 배열

## ✓ 배열을 사용하기 위해서는!

1. 배열의 선언
2. 배열의 생성
3. 배열의 사용

# 01 배열

## ✓ 배열의 선언

### Example

배열안에 저장되는  
자료형  `int[] data;`  배열의 이름

`/* elice */`

# 01 배열

## ✓ 배열의 선언

### Example

```
int[10] data;
```



배열 선언시에는 배열의 길이가 아직  
정해지지 않았으므로 표시하면 **안됨**

`/* elice */`

# 01 배열

## ✓ 배열의 생성

### Example

```
int[] data; //선언  
data = new int[3]; //생성
```

data	=	new	int	3
배열 이름		배열 생성을 위한 키워드	배열안에 저장되는 자료형	배열의 길이

/\* elice \*/

# 01 배열

## ✓ 배열의 생성

배열을 생성하게 되면  
배열 안에 저장되는 자료형에 따라  
**초기값(default value)**이 들어있다.

# 01 배열

## ✓ 배열의 초기값

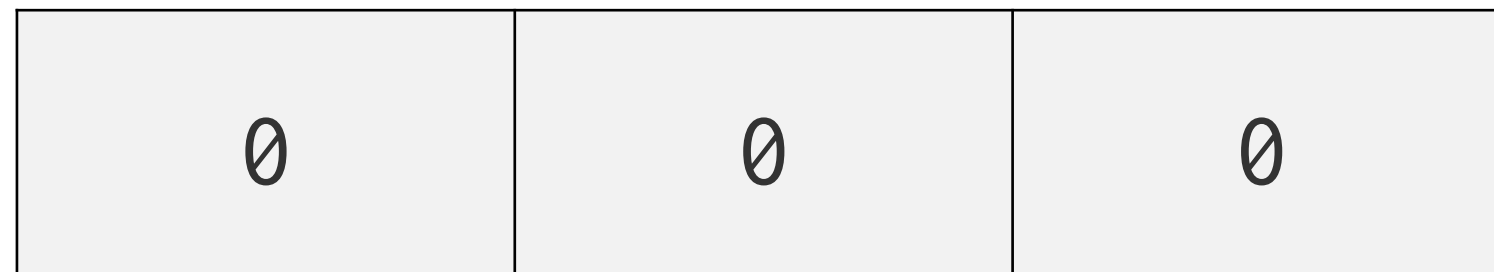
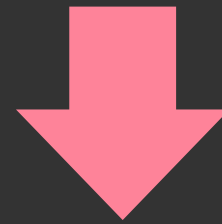
배열 안 자료형	초기값
정수형	0
실수형	0.0
논리형	false
문자열	null

# 01 배열

## ✓ 배열의 생성

### Example

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



/\* elice \*/

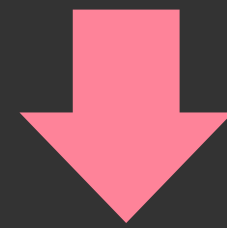


# 01 배열

## ✓ 배열의 인덱스

### Example

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



0	0	0
---	---	---

data[0] data[1] data[2]

/\* elice \*/

# 01 배열

## ✓ 배열의 인덱스

배열의 인덱스는 0부터 시작한다.

# 01 배열

## 배열의 생성

### Example

```
double[] points;  
points = new double[100];
```

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0
[0]	[1]	[2]			...				[99]

/\* elice \*/

# 01 배열

## ✓ 배열의 생성

### Example

```
String[] message;  
message = new String[5];
```

null	null	null	null	null
[0]	[1]	[2]	[3]	[4]

/\* elice \*/

# 01 배열

## ✓ 배열의 사용

### Example

```
int[] data;  
data = new int[3];  
data[0] = 123;
```

data

[0]

=

123

배열 이름

배열 안에서의 위치(index)

배열 안에 넣을 값

/\* elice \*/

# 01 배열

## ✓ 정수형 배열의 사용

### Example

```
int[] data;  
data = new int[3];  
data[0] = 123;  
data[1] = 345;  
data[2] = 567;
```

/\* elice \*/

# 01 배열

## ✓ 실수형 배열의 사용

### Example

```
double[] points;  
points = new double[100];  
points[0] = 3.12;  
points[5] = 6.19;  
...  
points[99] = 78.5;
```

/\* elice \*/

# 01 배열

## ✓ 문자열 배열의 사용

### Example

```
String[ ] messages;  
messages = new String[5];  
messages[0] = "hello, tommy!";  
messages[2] = "hi, jane!";  
messages[4] = "bye, harry!";
```

/\* elice \*/



# 01 배열

## ✓ 배열의 길이

배열 안에 저장할 수 있는 **데이터 개수**

Example

```
배열이름.length;
```

`/* elice */`

# 01 배열

## ✓ 배열의 길이

### Example

```
int[] scores = new int[3];  
scores[0] = 28;  
scores[1] = 64;  
scores[2] = 75;
```

scores

28	64	75
[0]	[1]	[2]

scores.length = 3

/\* elice \*/

# 01 배열

## ✓ 배열의 길이와 index

### Example

```
int[] scores = new int[170];
```

### Example

```
scores의 마지막 index == scores.length - 1
```

/\* elice \*/

# 01 배열

## ✓ 다른 방법으로 배열 사용하기

배열의 선언, 생성, 초기화를  
한번에!

`/* elice */`

# 01 배열

## ✓ 배열의 초기화 선언

### Example

```
int[] scores = {100, 45, 67, 88, 94};
```

```
double[] nums = {2.3, 4.5, 6.7, 9.0, 3.9};
```

```
boolean[] flags = {true, false, false};
```

```
char[] signs = {'q', 'w', 'e', 'r', 't', 'y'};
```

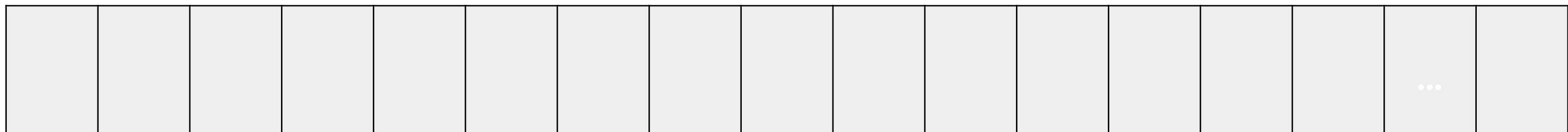
```
/* elice */
```

# 01 배열

## ✓ 배열의 초기화 선언

### Example

```
int[] data = new int[500];
```



[0] [1] [2] ...

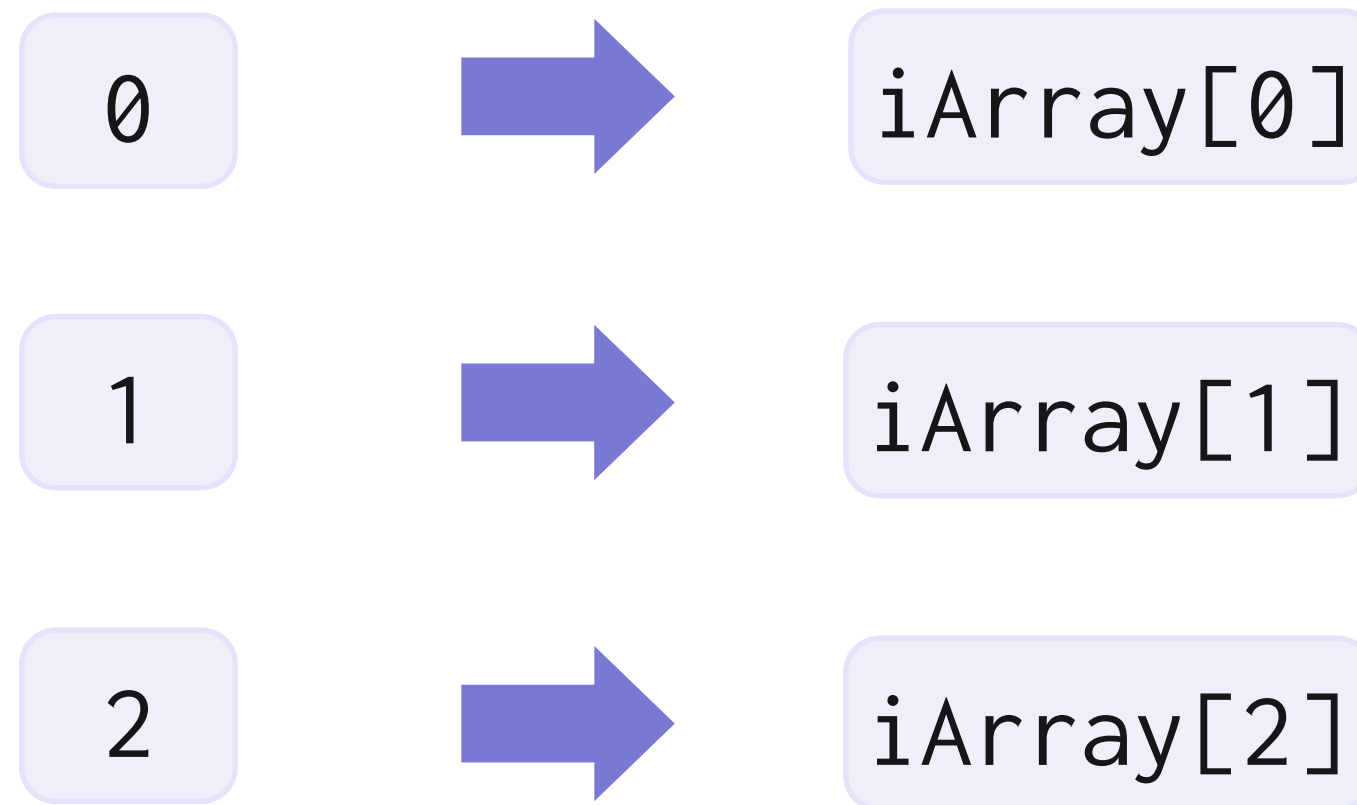
[499]

/\* elice \*/

# 01 배열

## ✓ [실습1] 배열 생성과 출력

배열을 생성하여 값을 넣어봅시다!



`/* elice */`

# 01 배열

## ✓ [실습2] 배열의 길이 확인하기

`length` 를 사용하여 **배열의 길이**를 출력해봅시다!

`/* elice */`



02

# 배열과 반복문



## 02 배열과 반복문

### ✓ 배열 안의 데이터 처리

배열안의 **모든** 데이터를 출력하라!

```
/* elice */
```

## 02 배열과 반복문

### ✓ 배열 안의 데이터 처리

#### Example

```
System.out.println(data[0]);  
System.out.println(data[1]);  
.  
.  
System.out.println(data[488]);  
System.out.println(data[499]);
```

/\* elice \*/

## 02 배열과 반복문

### ✓ 배열 안의 데이터 처리

#### Example

```
int[] data = new int[500];  
  
for (int i = 0; i < 500; i++) {  
    System.out.println(data[i]);  
}
```

/\* elice \*/

## 02 배열과 반복문

### ✓ 배열 안의 데이터 처리

#### Example

```
int[] data = new int[500];  
  
for (int i = 0; i < data.length; i++) {  
    System.out.println(data[i]);  
}
```

/\* elice \*/

## 02 배열과 반복문

### ✔ 향상된 for문(for each 문)

#### Example

```
int[] data = new int[500];  
  
for (int x : data) {  
    System.out.println(x);  
}
```

/\* elice \*/

## 02 배열과 반복문

### ✓ 배열 안의 데이터 처리

#### Example

```
String[] names = {"tommy", "jane", "bill"};

for (String x : names) {
    System.out.println(x);
}
```

/\* elice \*/

## 02 배열과 반복문

### ✓ [실습3] 반복문으로 배열 사용하기

for 문을 사용하여 모든 배열의 요소를 출력해봅시다!

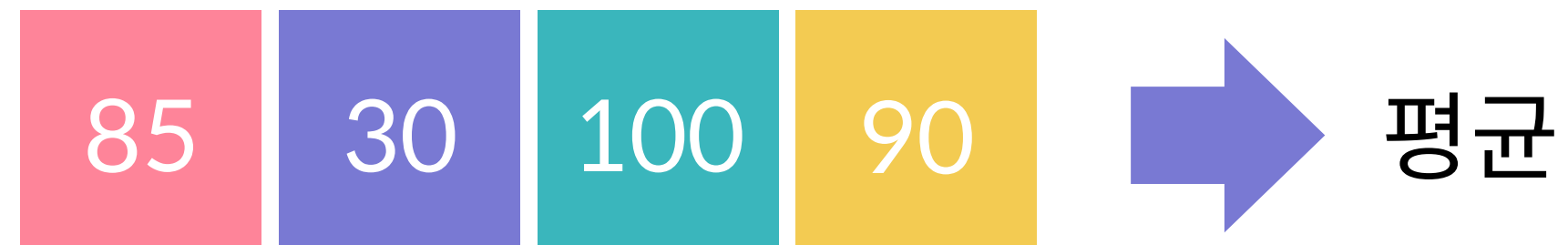
/\* elice \*/



## 02 배열과 반복문

### ✓ [실습4] 평균 구하기

grades 배열의 들어있는 모든 값을 더해 **평균**을 구해봅시다!

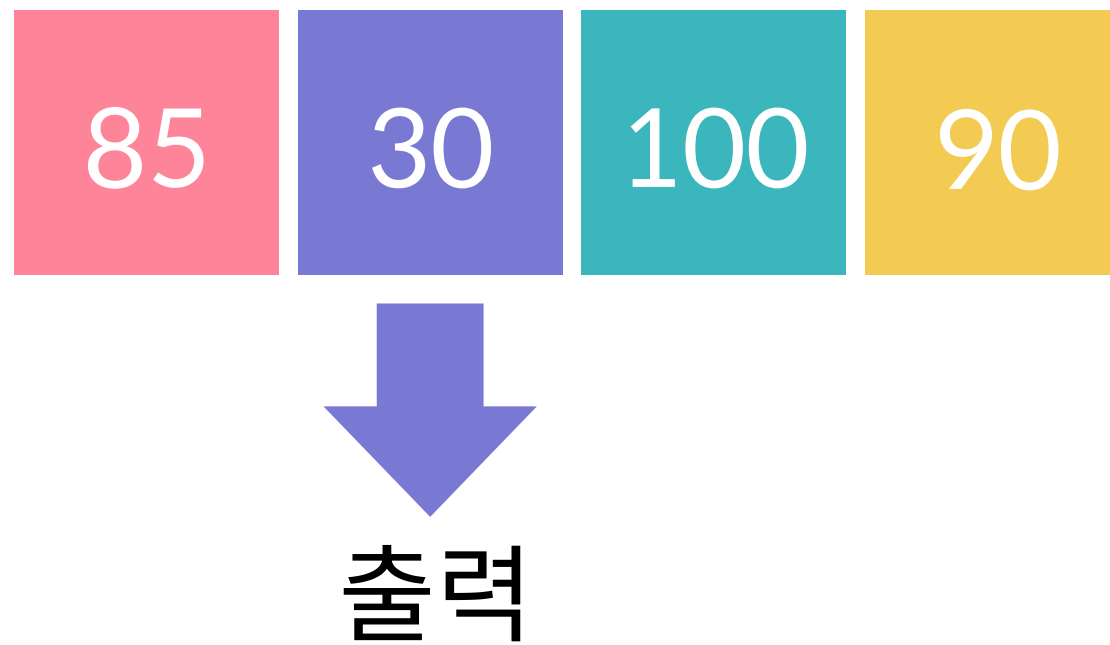


`/* elice */`

## 02 배열과 반복문

### ✓ [실습5] 첫 짝수 찾기

배열의 값을 차례로 검사해 처음으로 나오는 짝수를 찾아봅시다!



/\* elice \*/

## 02 배열과 반복문

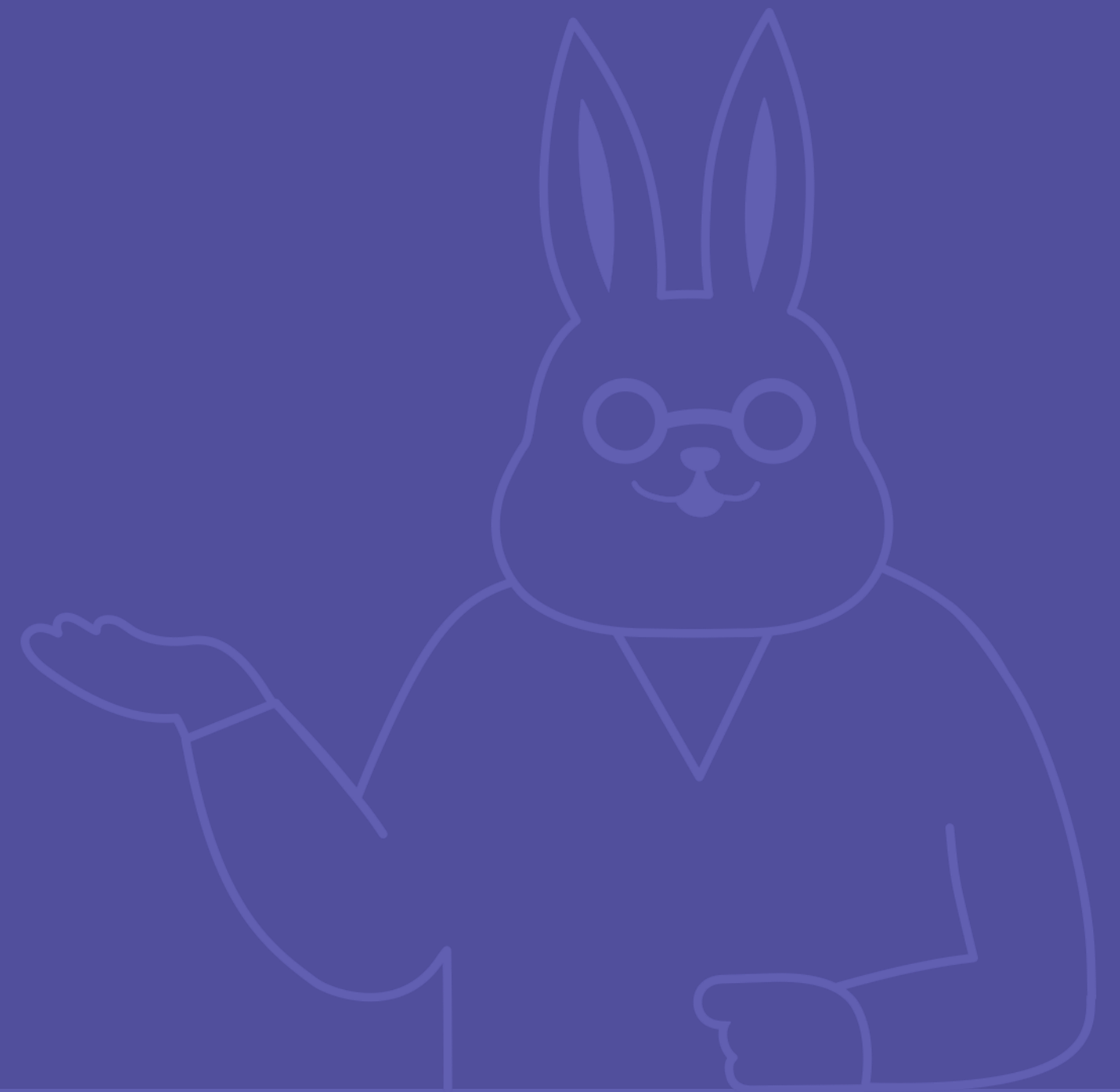
### ✓ [실습6] 향상된 for문

향상된 `for` 문을 사용하여 배열의 있는 **모든 값**을 출력해봅시다!

`/* elice */`

03

# 이차원 배열



## 03 이차원 배열

### ✓ 이차원 배열의 필요성

배열을 이용하여 **학생이 많은 경우**에도  
성적을 편리하게 관리

국어 성적



# 03 이차원 배열

## 이차원 배열의 필요성

수학 성적도 배열로 만들어봅시다.

국어 성적	85	30	100	90
수학 성적	70	100	40	65

## 03 이차원 배열

### ✓ 이차원 배열의 필요성

영어, 과학, 음악...  
계속 과목이 **추가**된다면?

국어 성적	85	30	100	90
수학 성적	70	100	40	65

# 03 이차원 배열

## 이차원 배열의 필요성

각 요소가 배열로 구성된 배열을 사용해봅시다!

전체 성적	[0]	85	30	100	90	국어
	[1]	70	100	40	65	수학
	[2]	85	30	100	90	영어
	[3]	70	100	40	65	과학



## 03 이차원 배열

### ✓ 이차원 배열이란?

각 요소가 **일차원 배열**인 배열

이차원 이상의 배열은 **다차원 배열**

# 03 이차원 배열

## 이차원 배열의 생성

### Example

```
int[][] data = {{1,2,3},{4,5,6}};
```

	data[0][0]	data[0][1]	data[0][2]
data[0]	1	2	3
data[1]	4	5	6
	data[1][0]	data[1][1]	data[1][2]

/\* elice \*/

## 03 이차원 배열

### ✔ 이차원 배열의 구조

	<code>data[0][0]</code>	<code>data[0][1]</code>	<code>data[0][2]</code>
<code>data[0]</code>	1	2	3
<code>data[1]</code>	4	5	6
	<code>data[1][0]</code>	<code>data[1][1]</code>	<code>data[1][2]</code>

#### Example

```
data.length == 2
```

`/* elice */`

## 03 이차원 배열

### ✔ 이차원 배열의 구조

	<code>data[0][1]</code>	<code>data[0][2]</code>	<code>data[0][3]</code>
<code>data[0]</code>	1	2	3
<code>data[1]</code>	4	5	6
	<code>data[1][0]</code>	<code>data[1][1]</code>	<code>data[1][2]</code>

#### Example

```
data[0].length == 3
```

`/* elice */`

## 03 이차원 배열

### ✓ 이차원 배열의 사용

#### Example

```
int[][] data = {{1,2,3}, {4,5,6}};

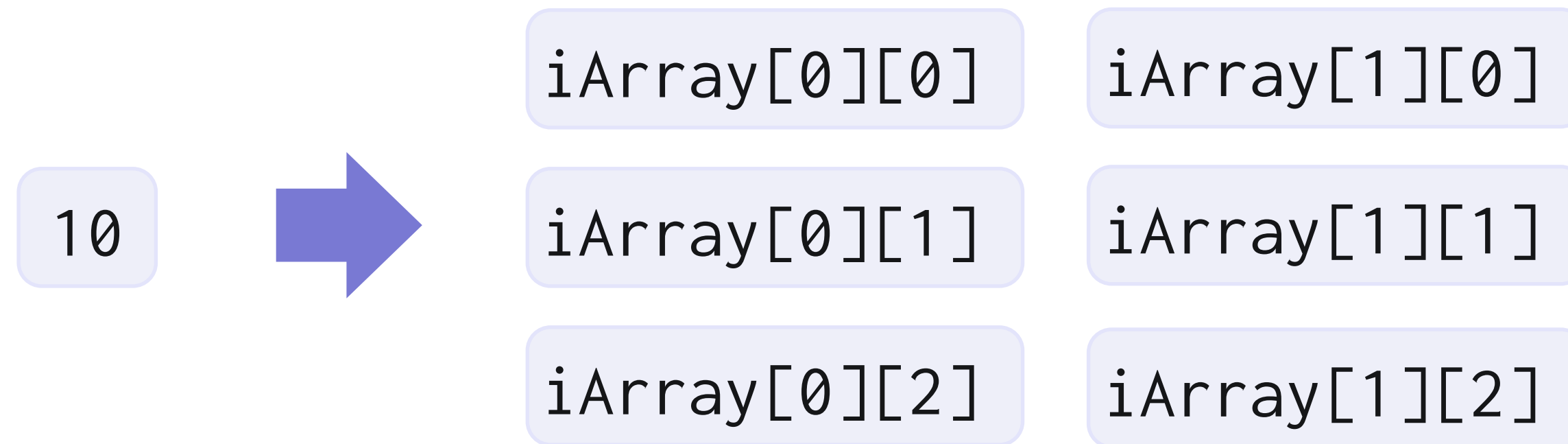
for(int i = 0; i < data.length; i++) {
    for(int j = 0; j < data[i].length; j++) {
        System.out.println(data[i][j]);
    }
}
```

/\* elice \*/

## 03 이차원 배열

### ✓ [실습7] 이차원 배열

이차원 배열 안에 **같은 값**을 모두 넣어 봅시다!



/\* elice \*/

## 03 이차원 배열

### ✓ [실습8] 이차원 배열 반복하기

**이차원 배열** 안에 있는 모든 값을 출력해봅시다!

`/* elice */`

# Credit

/\* elice \*/

코스 매니저

강윤수

콘텐츠 제작자

강윤수

강사

유동환 선생님

디자인

박주연



# Contact

TEL

070-4633-2015

WEB

<https://elice.io>

E-MAIL

[contact@elice.io](mailto:contact@elice.io)

