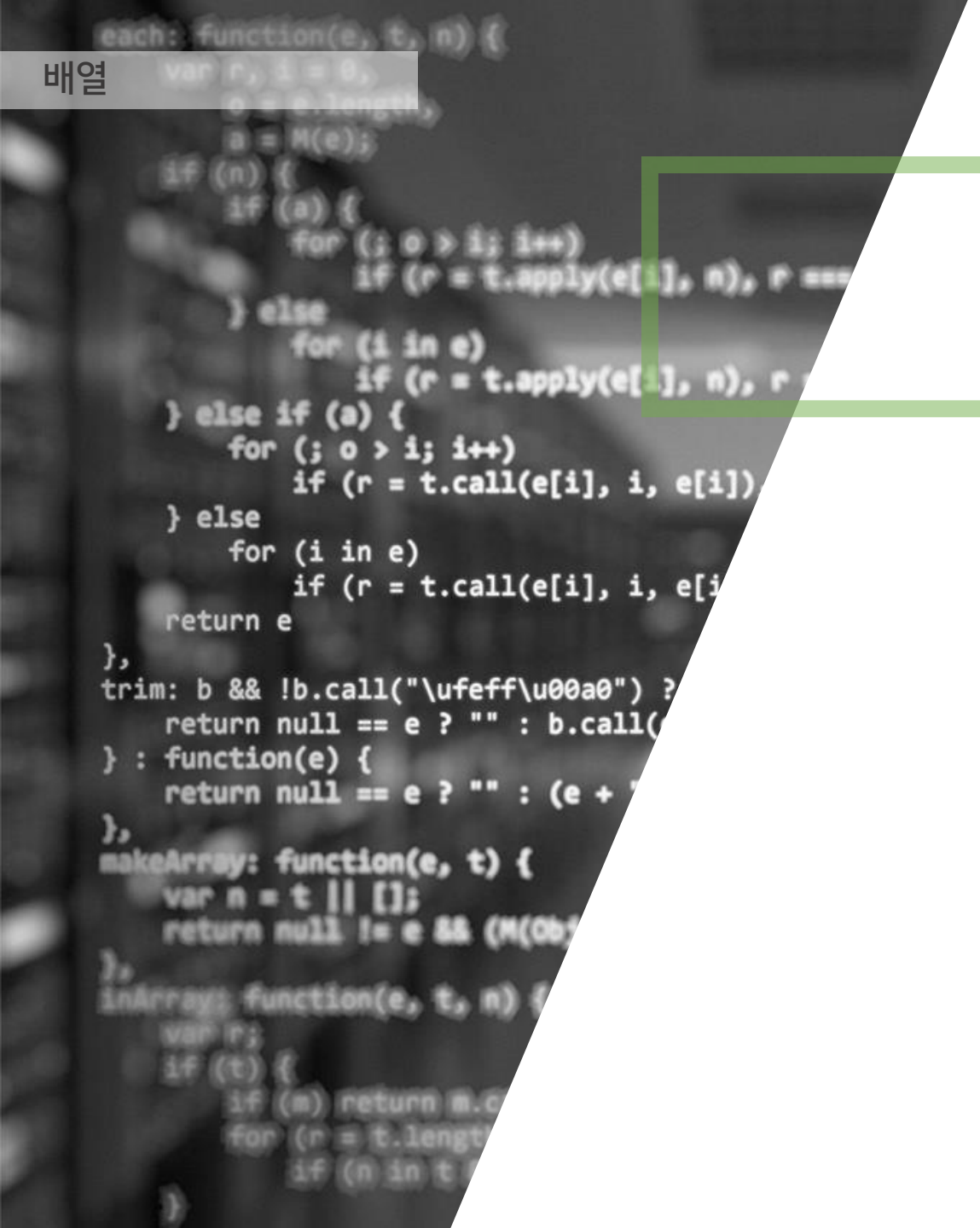
제05장

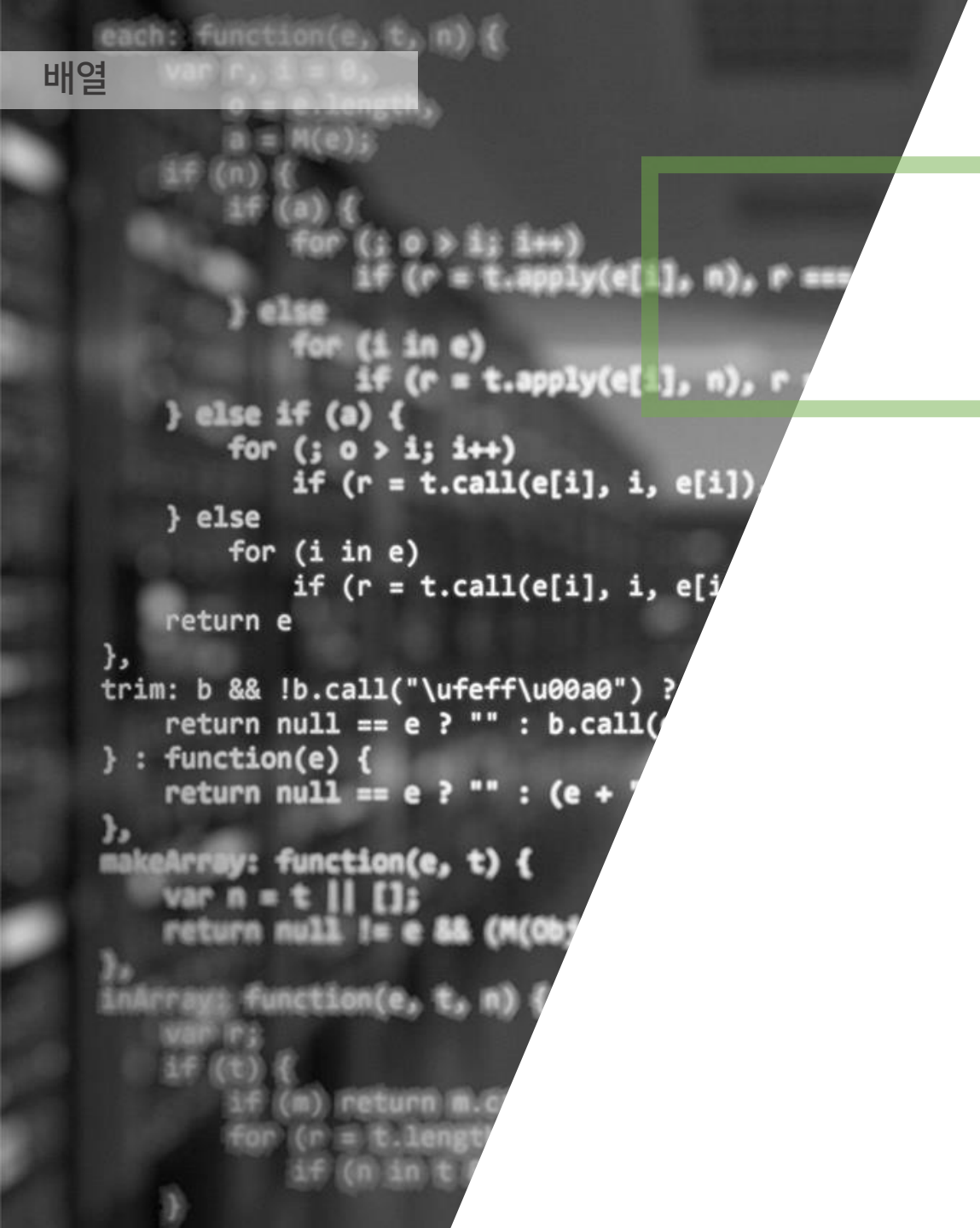
배열

구디아카데미 ▷ 민경태 강사

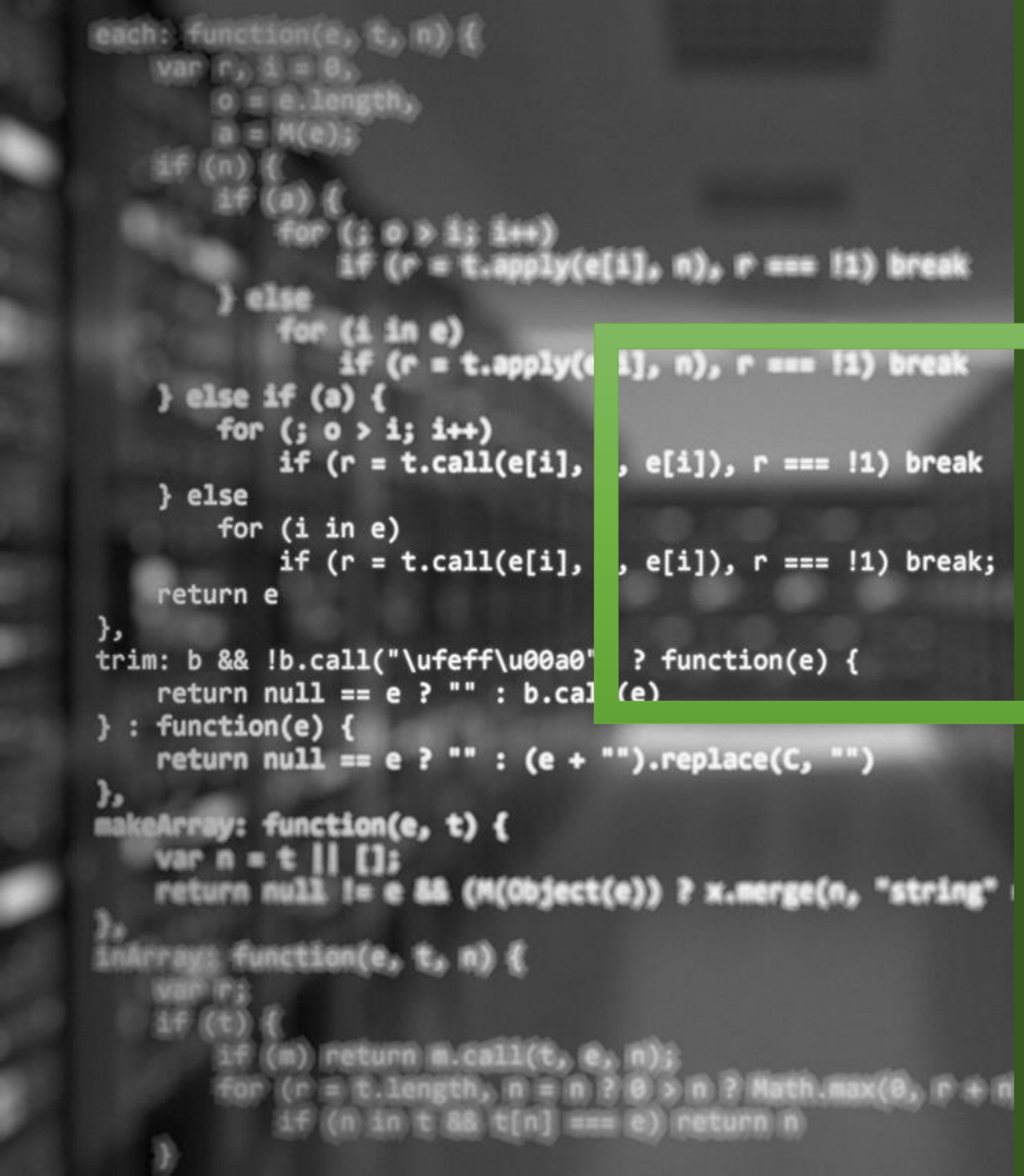


학습목표

1. 배열의 구조에 대해서 알 수 있다.
2. 배열의 선언과 생성 방법을 알 수 있다.
3. 인덱스와 배열 요소에 대해서 알 수 있다.
4. 반복문을 이용하여 배열을 순회할 수 있다.



목차


each: function(e, t, n) {
var r, i = 0,
o = e.length,
a = M(e);
if (n) {
if (a) {
for (; o > i; i++)
if (r = t.apply(e[i], n), r === !1) break
} else
for (i in e)
if (r = t.apply(e[i], n), r === !1) break
} else if (a) {
for (; o > i; i++)
if (r = t.call(e[i], e[i]), r === !1) break
} else
for (i in e)
if (r = t.call(e[i], e[i]), r === !1) break;
return e
},
trim: b && !b.call("\ufeff\u00a0") ? function(e) {
return null == e ? "" : b.call(e)
} : function(e) {
return null == e ? "" : (e + "").replace(C, "")
},
makeArray: function(e, t) {
var n = t || [];
return null != e && (M(Object(e)) ? x.merge(n, "string"
},
inArray: function(e, t, n) {
var r;
if (t) {
if (m) return m.call(t, e, n);
for (r = t.length, n = n ? 0 > n ? Math.max(0, r + n
if (n in t && t[n] === e) return n
}

1. 배열이란?

배열의 도입 필요성

- 상황

> 이번 SQLD 시험에 총 500명이 응시하였다. 500명의 점수를 500개의 변수에 저장한 뒤 하나씩 꺼내서 각각의 점수가 60점 이상인지 비교하는 if문을 이용해 응시자들의 합격 여부를 판단하고자 한다.

Q1. 500개의 변수 이름은 어떻게 결정할 것인가?

Q2. 점수 비교를 위해서 500개의 if문을 작성해야 하는가?

A. 500개의 값을 저장할 수 있는 **배열(Array)**로 쉽게 해결 가능하다.

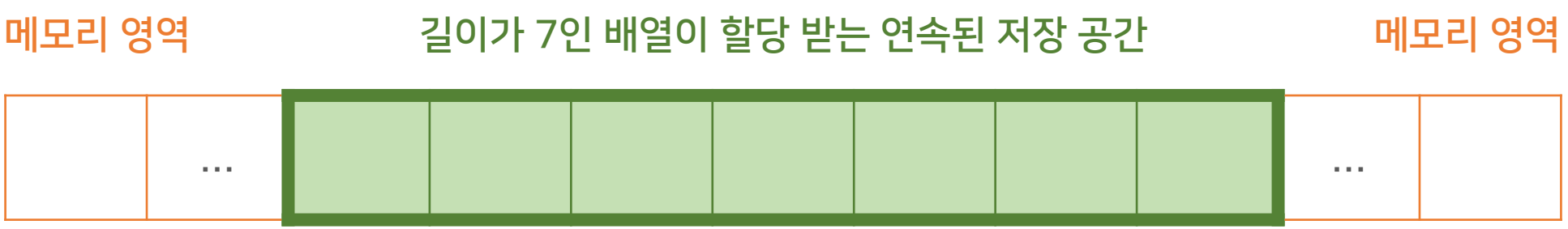
배열

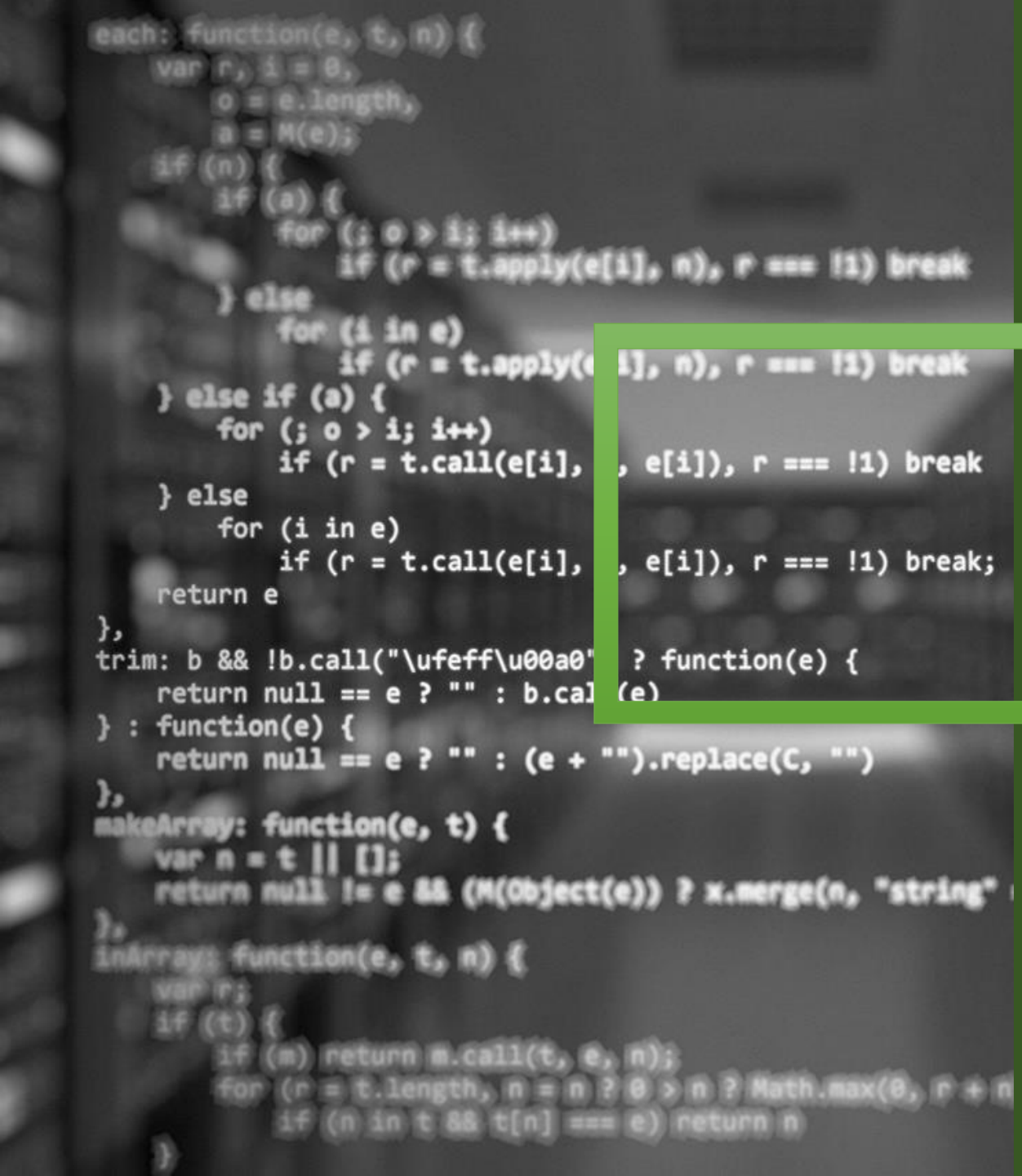
- Array
- 여러 개의 값을 저장할 수 있는 자료구조
- 동일한 타입을 가진 값들은 모두 하나의 배열에 저장할 수 있음
  - 서로 다른 타입을 가진 값을 하나의 배열에 저장할 수는 없음
- 배열에 저장된 모든 값들은 반복문을 이용해서 쉽게 처리할 수 있음

배열의 메모리 구조

- 배열은 메모리를 할당 받을 때 연속된 공간을 할당 받음

- 모든 데이터가 연속된 공간에 있기 때문에 빠르게 접근할 수 있음





2. 배열의 선언과 생성

배열의 선언

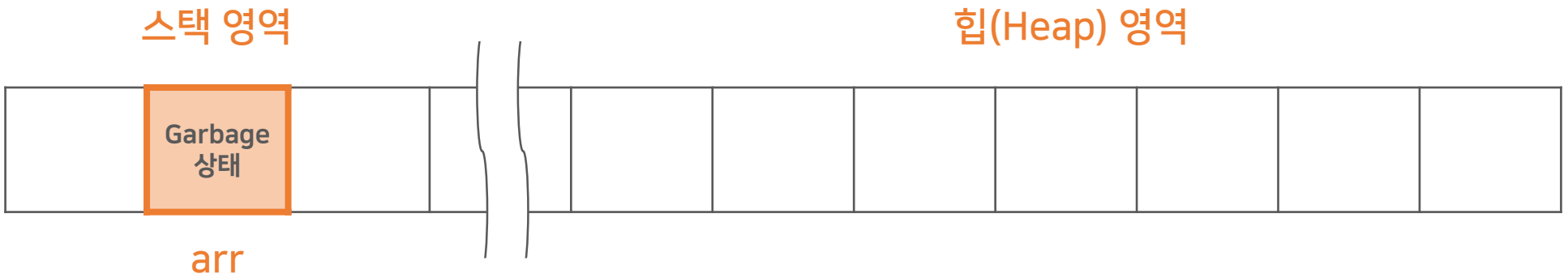
- 배열을 만들기 위해서 필요한 첫 번째 단계임
- 배열의 선언이 배열의 생성을 의미하지는 않으므로 주의해야 함
- 배열을 선언한 뒤 배열 생성 과정을 추가로 거쳐야 배열이 만들어 짐
- 배열명은 참조 변수임(배열의 참조값을 가지는 변수)
- 형식
  1. 자료형[] 배열명 (추천하는 방식)
  2. 자료형 배열명[]

배열의 선언

- 배열을 선언할 때는 배열의 길이를 정하지 않음

- 예시

```
int[] arr;
```



arr라는 이름을 가진 참조 변수가 생긴다.

배열의 생성

- 배열의 생성은 연속된 메모리를 할당 받는 과정을 의미함
- 힙(Heap) 영역의 메모리를 할당 받음
- 생성할 배열의 길이를 결정해야 함
- 배열 선언 시 생성된 참조 변수(배열명)에 할당 받은 메모리의 주소값(참조값)을 저장하는 방식으로 배열을 생성함
- 형식

  배열명 = new 자료형[배열길이]

배열의 생성

- 배열이 생성되면 배열의 모든 요소들은 자동으로 초기화가 됨

- 배열 요소의 자료형에 따른 자동 초기화 값

유형	자료형	값
기본 자료형	정수형	0
	실수형	0.0
	논리형	false
	문자형	₩u0000 (null문자)
참조 자료형		null

배열의 길이

- 배열을 생성할 때 배열의 길이를 정해야 함

- 한 번 생성된 배열의 길이는 수정할 수 없음
  - 배열의 길이를 수정하려면 새로운 배열을 만들고 기존 배열의 모든 요소를 옮기는 방법을 사용해야 함

- 형식

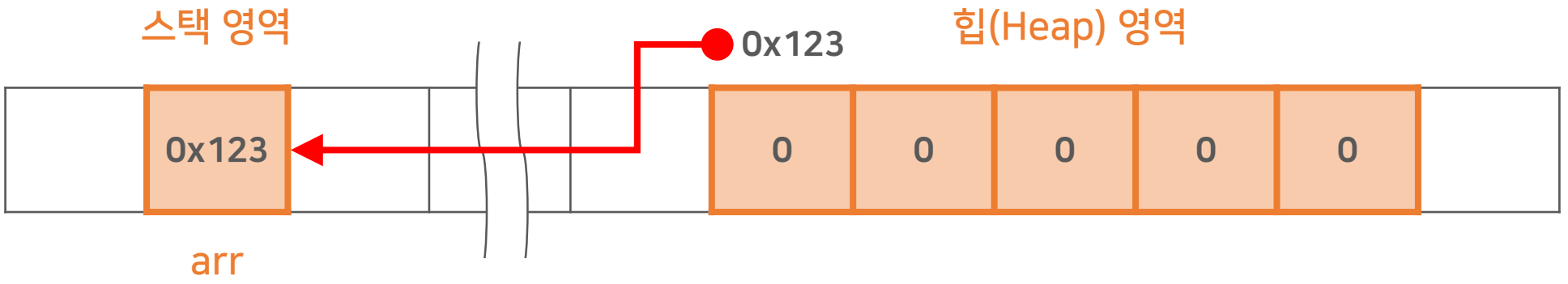
  배열명.length

배열의 생성

- new 키워드를 이용해서 Heap 영역의 메모리를 할당 받음

- 예시

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



배열의 초기화

- 배열을 생성하면서 각 요소의 값을 원하는 값으로 초기화 할 수 있음
- 배열 요소에 일일이 접근하면서 값을 저장할 필요가 없으므로 편리함
- 형식
  - 자료형[] 배열명 = {값1, 값2, 값3, ...}
  - 자료형[] 배열명 = new 자료형[] {값1, 값2, 값3, ...}

배열의 초기화

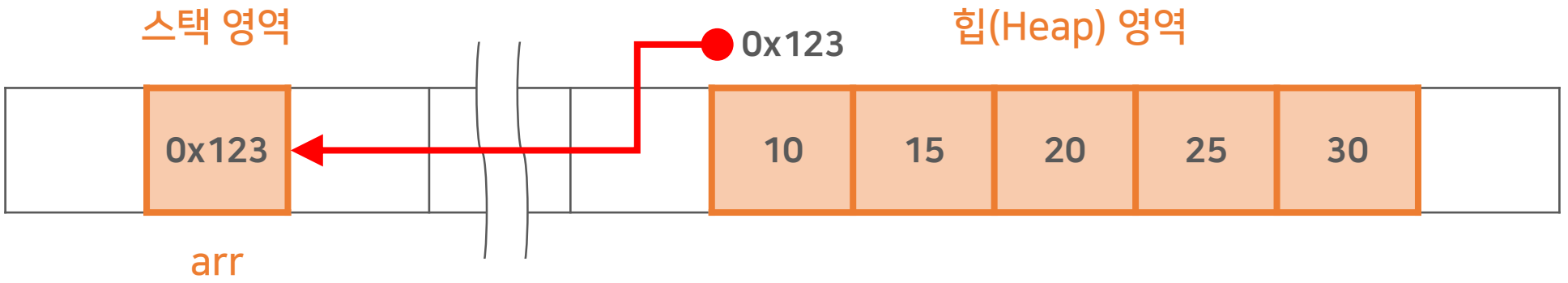
- 배열을 초기화 할 때는 중괄호{}를 사용함

- 예시

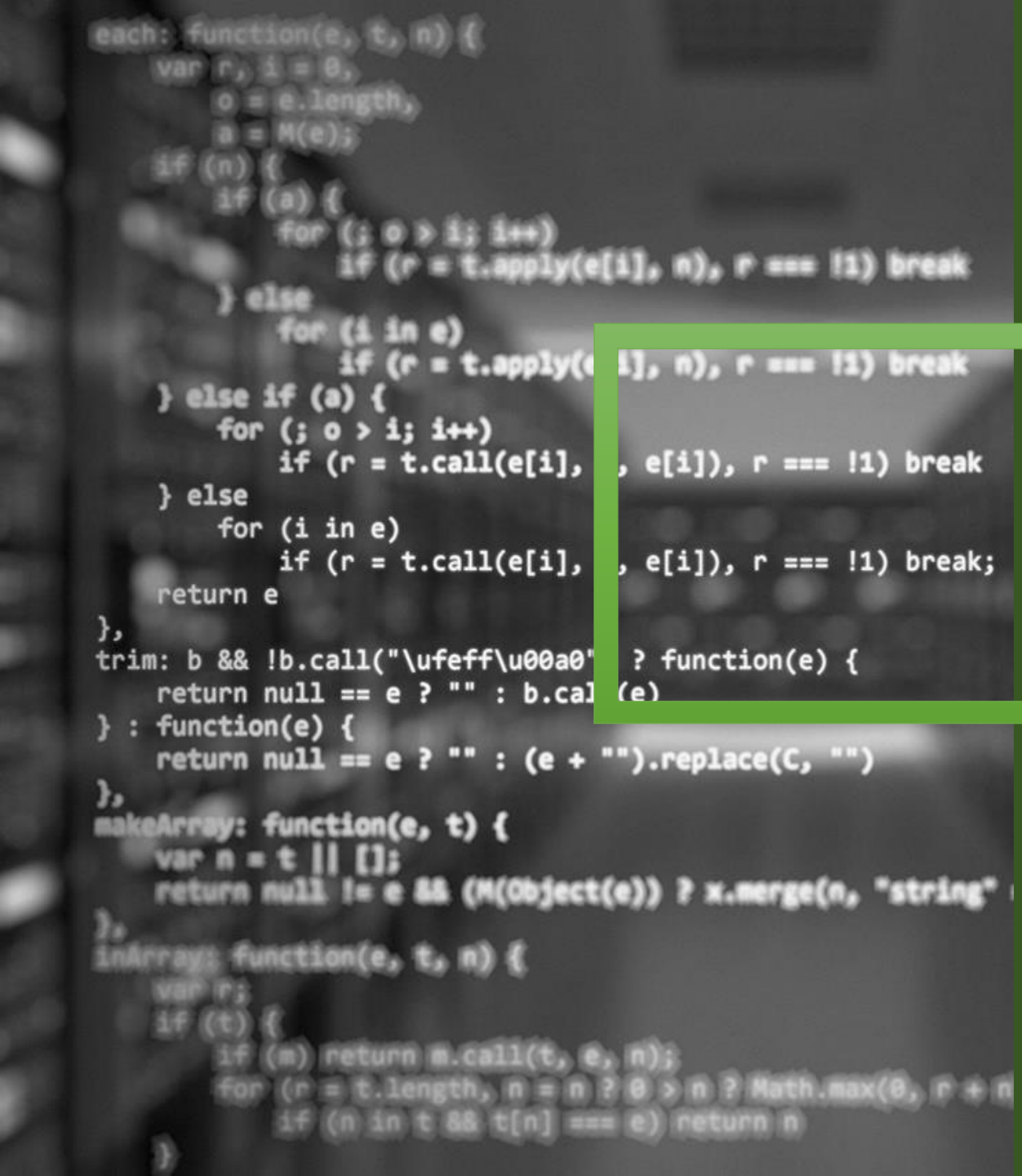
```
int[] arr = {10, 15, 20, 25, 30};
```

또는

```
int[] arr = new int[] {10, 15, 20, 25, 30};
```



중괄호{}에 작성한 순서대로 초기화된다.



3. 인덱스와 배열 요소

인덱스와 배열 요소

- 인덱스(Index)
  - 배열 요소를 구분하기 위한 고유의 숫자
  - 0 ~ (배열의 길이 - 1) 사이*의 값을 가질 수 있음

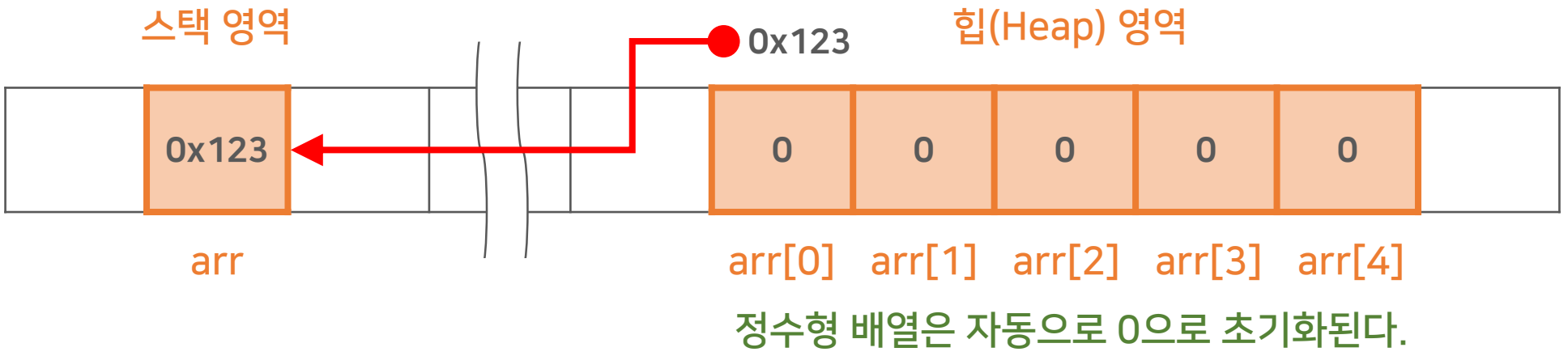
- 배열 요소(Element)
  - 배열에 저장된 각각의 데이터를 의미함
  - 대괄호[]와 인덱스를 이용해서 배열 요소를 나타냄

* 범위를 벗어난 인덱스를 사용하면 ArrayIndexOutOfBoundsException이라는 예외가 발생한다.
* 예외는 [11.예외처리]에서 다룬다.

배열 요소

- 배열 요소 형식
  배열명[인덱스]

- 예시
  ```
  int[] arr;
  arr = new int[5];
  ```

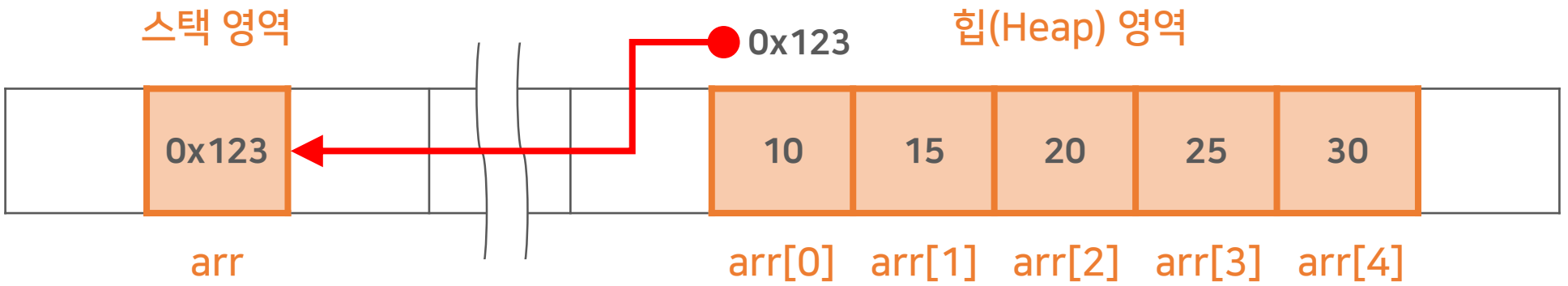


배열 요소와 인덱스

- 배열 요소는 각각이 하나의 변수임

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

arr[0] = 10;  arr[1] = 15;  arr[2] = 20;  arr[3] = 25;  arr[4] = 30;
```



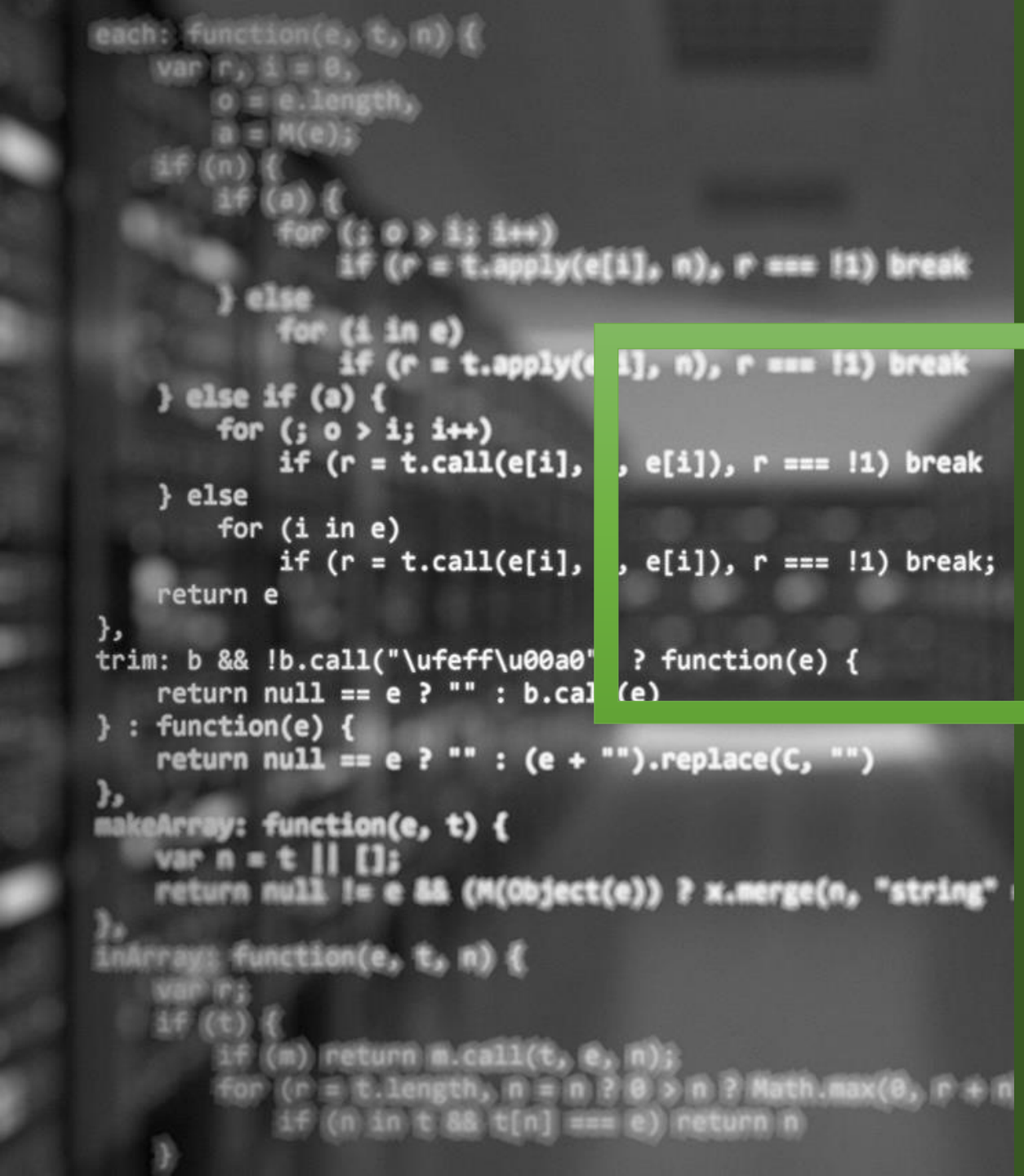
배열 요소와 인덱스 예시

- {"봄", "여름", "가을", "겨울"}로 초기화 된 String[] seasons 배열

```
public class MainClass {
    public static void main(String[] args) {
        String[] seasons = {"봄", "여름", "가을", "겨울"};
        System.out.println(seasons[0]);
        System.out.println(seasons[1]);
        System.out.println(seasons[2]);
        System.out.println(seasons[3]);
    }
}
```

실행결과

```
"봄"
"여름"
"가을"
"겨울"
```



4. 반복문을 이용한 배열의 순회

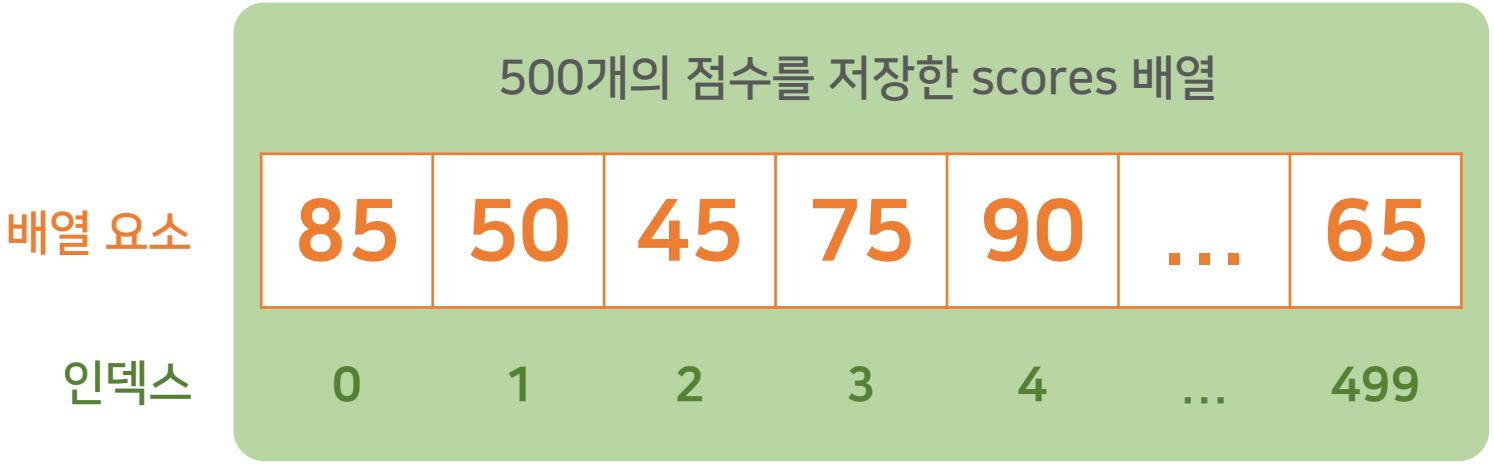
for문과 배열

- 배열은 0부터 1씩 증가하는 인덱스를 이용해서 배열 요소를 구분함
- 인덱스를 기준으로 반복문을 작성하면 배열 요소를 순회할 수 있음
- 일반적으로 배열 순회 시 사용하는 반복문은 for문임
- 관례상 인덱스의 변수 이름은 i를 사용함

for문과 배열 예시

- 500명의 점수를 가지고 있는 scores 배열을 순회하면서 합격 여부 출력

```
public class MainClass {
    public static void main(String[] args) {
        int[] scores = {85, 50, 45, 75, 90, ... 65};
        for(int i = 0; i < scores.length; i++) {
            if(scores[i] >= 60) {
                System.out.println((i + 1) + "번 " + scores[i] + "점 합격");
            }
        }
    }
}
```



향상 for문

- Advanced For

- 인덱스를 사용하지 않고 배열을 순회할 수 있는 for문임

- 배열의 모든 요소를 알아서 순서대로 가져오는 for문임

- 형식

```
for(변수 선언 : 배열명) {

}
```

향상 for문과 배열 예시

- 500명의 점수를 가지고 있는 scores 배열을 순회하면서 합격 여부 출력

```
public class MainClass {
    public static void main(String[] args) {
        int[] scores = {85, 50, 45, 75, 90, ... 65};
        for(int score : scores) {
            if(score >= 60) {
                System.out.println(score + "점 합격");
            }
        }
    }
}
```