

# 프로그래머스 - 배열 자르기

≡ 태그

IntStream.rangeClosed

numbers.slice

자바스크립트

배열 자르기

문제 설명

정수 배열 numbers와 정수 num1, num2가 매개변수로 주어질 때, numbers의 num1번째 인덱스부터 num2번째 인덱스까지 자른 정수 배열을 return 하도록 solution 함수를 완성해보세요.

제한사항

$2 \leq \text{numbers의 길이} \leq 30$

$0 \leq \text{numbers의 원소} \leq 1,000$

$0 \leq \text{num1} < \text{num2} < \text{numbers의 길이}$

입출력 예 #1

numbers: [1, 2, 3, 4, 5] , num1: 1, num2: 3 , result: [2, 3, 4]

[1, 2, 3, 4, 5]의 1번째 인덱스 2부터 3번째 인덱스 4 까지 자른 [2, 3, 4]를 return 합니다.

입출력 예 #2

numbers: [1, 3, 5] , num1: 1, num2: 2 , result: [3, 5]

[1, 3, 5]의 1번째 인덱스 3부터 2번째 인덱스 5까지 자른 [3, 5]를 return 합니다.

```
function solution(numbers, num1, num2) {  
    return numbers.slice(num1, num2 + 1);  
}
```

slice 메서드는 시작 인덱스와 끝 인덱스라는 두 개의 인수를 사용합니다. 끝 인덱스는 배타적이므로 결과에 해당 인덱스에 있는 요소를 포함하려면 1씩 증가해야 합니다.

numbers.slice(num1, num2 + 1) 문은 슬라이스 메서드를 사용하여 입력 배열 "numbers"의 하위 배열인 새 배열을 반환합니다.

슬라이스 메서드는 시작 인덱스와 끝 인덱스라는 두 개의 인수를 사용합니다. 시작 인덱스는 num1이고 종료 인덱스는 num2+1입니다. 끝 인덱스는 배타적이므로 결과에 해당 인덱스에 있는 요소를 포함하려면 1씩 증가해야 합니다. 슬라이스 방법은 새 배열을 반환하며 원래 배열을 수정하지 않습니다.

예를 들어 입력 배열이 [1, 2, 3, 4, 5]이고 시작 인덱스가 1이고 종료 인덱스가 3인 경우 문 numbers.slice(num1, num2 + 1)은 [2, 3, 4]

따라서 numbers.slice(num1, num2 + 1) 문은 인덱스 "num1"에서 시작하여 인덱스 "num2"에서 끝나는 입력 배열 "numbers"의 하위 배열인 새 배열을 반환하는 데 사용됩니다. .

```
import java.util.stream.IntStream;

class Solution {
    public int[] solution(int[] numbers, int num1, int num2) {
        return IntStream.rangeClosed(num1, num2).map(i -> numbers[i]).toArray();
    }
}
```

귀하가 제공한 Java 솔루션은 `java.util.stream` 패키지의 `IntStream` 클래스를 사용하여 `num1`에서 `num2` (포함) 범위의 정수 스트림을 생성합니다. `rangeClosed` 메서드는 `num1`에서 `num2`(포함)까지의 정수 스트림을 만들고 `map` 메서드는 스트림의 각 요소에 함수를 적용합니다. 이 경우 함수는 정수 스트림을 다음으로 매핑하는 `i -> numbers[i]`입니다. 숫자 배열의 해당 요소. 마지막으로 `toArray()` 메서드는 정수 스트림을 배열로 변환하는 데 사용됩니다.

솔루션은 문제를 해결하기 위해 기능적 접근 방식을 사용하고 있으며 이전 솔루션보다 더 간결하고 효율적입니다. 그러나 이 솔루션은 `num1` 및 `num2`가 숫자 배열의 유효한 인덱스이고 `num1`이 `num2`보다 작다고 가정합니다. `ArrayIndexOutOfBoundsException`을 피하기 위해 솔루션을 실행하기 전에 이러한 가정이 참인지 확인하는 것이 중요합니다.