

4. [Java] ??? ??? - return new String(arr);

<https://school.programmers.co.kr/learn/courses/30/lessons/120895>

□□

```
class Solution {  
    public String solution(String my_string, int num1, int num2) {  
        String[] arr = String.toCharArray(my_string);  
        char tmp = arr[num1];  
        arr[num1] = arr[num2];  
        arr[num2] = tmp;  
        return arr.toString();  
    }  
}
```

? String[] arr = String.toCharArray(my_string);

- → String.toCharArray() □ char[] □ □□□□ □□□ String[] □ □□ char[]□□□ □□ .

? String.toCharArray() ??

- → my_string.toCharArray();
- toCharArray() □ String □□□□ □□□□ □□□□ .
- □□□ String.toCharArray() □ (□□□ □□□ □□)
- □□□ □□□□ (my_string) □ □□□ □□ .

? return arr.toString();

- char[]□□ toString() □ □□ □□□□ □□□□ .
- → □□ new String(arr) □ □□□□ □□□□ □□□□ .

□□□ □□

```
class Solution {
    public String solution(String my_string, int num1, int num2) {
        char[] arr = my_string.toCharArray();
        char tmp = arr[num1];
        arr[num1] = arr[num2];
        arr[num2] = tmp;
        return new String(arr);
    }
}
```

??? VS ????

`uint8_t` `uint16_t` `int` `int16_t` `uint32_t` `**uint8_t` `uint8_t` `(static uint8_t)` `**uint8_t` `, int` `uint8_t`

`uint8_t` `uint16_t` `int` `uint32_t` `int` `uint8_t` `uint16_t` `uint32_t` .

```

// static method to parse a string into an integer
// Example: Integer.parseInt("123") returns 123
// Note: This method throws a NumberFormatException if the string is not a valid integer.

```

1. 在 `main` 函数中，定义一个 `String` 类型的变量 `str`，并赋值为 `"Hello, World!"`。
 2. 使用 `str.length()` 方法获取字符串的长度，并存储在变量 `len` 中。
 3. 使用 `str.substring(0, len)` 方法获取字符串的副本，并存储在变量 `str2` 中。
 4. 使用 `str2` 变量输出字符串。

```

class Person {
    name: String
    sayHello(): Unit {
        println("Hello, my name is " + name)
    }
}

val person1 = Person("Alice")
val person2 = Person("Bob")

person1.sayHello()
person2.sayHello()

```

0000 0000 0000 000 000 0000 000 000 0 0000 , 000 0000 00 00
 00 000 000 0 0000 . 0000 000 000 00 0 000 0 000 00 0000 00 .
 0000 0000 000 00000 0000 000 , 00 00 0000 00 0000 00000
 000 0000 0000 00 00 .

? 1. ???? ??? (Instance Method)

[illegible]

- `new` 키워드 없이도 객체를 생성할 수 있다
- `new` 키워드 없이도 객체를 생성할 수 있다

예제

- `new` 키워드 없이도 객체를 생성할 수 있다
- `new` 키워드 없이도 객체를 생성할 수 있다

예제

```
public class Person {
    String name;

    public void sayHello() {
        System.out.println("Hello, my name is " + name);
    }
}

// 실행
Person p = new Person();
p.name = "Dain";
p.sayHello(); // 출력: Hello, my name is Dain
```

? 2. ??? ??? (Class Method)

예제

- `static` 키워드 없이도 메서드를 호출할 수 있다
- `static` 키워드 없이도 메서드를 호출할 수 있다

예제

- `static` 키워드 없이도 메서드를 호출할 수 있다
- `static` 키워드 없이도 메서드를 호출할 수 있다
- `static` 키워드 없이도 메서드를 호출할 수 있다

예제

```
public class MathUtil {
    public static int add(int a, int b) {
        return a + b;
    }
}
```

```
}

// 計算
int result = MathUtil.add(3, 5); // 3 + 5 = 8
```

Revision #17

Created 19 May 2025 00:40:16 by Dain

Updated 13 July 2025 14:23:32 by Dain