

???

- [1. \[Java\] 문자열 인덱싱 - cipher.charAt\(i - 1\)](#)
- [2. \[Java\] 문자열 길이 구하기 - str.length\(\)](#)
- [3. \[Java\] 문자열 대체 - numbers.replaceAll\(a,b\);](#)
- [4. \[Java\] 문자열 배열 생성 - return new String\(arr\);](#)
- [5. \[Java\] 문자열 빌더 사용하기 - sb.toString\(\);](#)
- [10. \[Java\] 문자열 찾기 - String.indexOf\(\), String.valueOf\(\)](#)

1. [Java] ?? ?? - cipher.charAt(i - 1)

<https://school.programmers.co.kr/learn/courses/30/lessons/120892>

??

```
class Solution {
    public String solution(String cipher, int code) {
        StringBuilder sb = new StringBuilder();
        for (int i = code; i <= cipher.length(); code++;) {
            sb.append(cipher[i-1]); // ????? 0?? ??
        }
        return sb.toString();
    }
}
```

? for?? ??? ??

```
for (int i = code; i <= cipher.length(); code++;)
```

- → code++ → ??? ?? ?? ? i(?? ?? ??)
- → code++ ? code ?? ?? ????? ????? ??

? ??? ??? ?? ??

- Java ?? ??? [] ?? ?? → charAt(index) ??? ?

?? ??

```
class Solution {  
    public String solution(String cipher, int code) {  
        StringBuilder sb = new StringBuilder();  
        for (int i = code; i <= cipher.length(); i += code) {  
            sb.append(cipher.charAt(i - 1)); // 문자를 0부터 시작  
        }  
        return sb.toString();  
    }  
}
```

2. [Java] ????? ???

<https://school.programmers.co.kr/learn/courses/30/lessons/120893>



```
class Solution {
    public String solution(String my_string) {
        StringBuilder sb = new StringBuilder();
        for(int i = 0; i < my_string.length(); i++) {
            if(Character.isUpperCase(my_string.charAt(i))) {
                sb.append(Character.toLowerCase(my_string.charAt(i)));
            } else {
                sb.append(Character.toUpperCase(my_string.charAt(i)));
            }
        }
        return sb.toString();
    }
}
```

my_string.charAt(i) char c = my_string.charAt(i); .



```
class Solution {
    public String solution(String my_string) {
        StringBuilder sb = new StringBuilder();
        for(int i = 0; i < my_string.length(); i++) {
            char c = my_string.charAt(i);
            if(Character.isUpperCase(c)) {
                sb.append(Character.toLowerCase(c));
            } else {
                sb.append(Character.toUpperCase(c));
            }
        }
    }
}
```

```
    }  
    return sb.toString();  
}  
}
```

3. [Java] ??? ??? - numbers.replaceAll(a,b);

<https://school.programmers.co.kr/learn/courses/30/lessons/120894>

?? ??

```
class Solution {  
    public long solution(String numbers) {  
        String[] words = {  
            "zero", "one", "two", "three", "four", "five",  
            "six", "seven", "eight", "nine"  
        };  
        for (int i = 0; i < words.length; i++) {  
            numbers = numbers.replace(words[i], String.valueOf(i));  
        }  
        return Integer.parseInt(numbers);  
    }  
}
```

???? ???? ?? ?? ?? ??

[image.png](#)

[image.png](#)

? Integer.parseInt() ? ? ? ? ? ?

- ?? ?? ???? int? ?? ???? ?? ???? .
- ??? ?? ?? ???? ?? ? ?
- ?? ?? , ?? 13?? ?? ???? int(32?? ??)? ?? 2,147,483,647? ??
- → Integer.parseInt()? ?? ? ? → ??? ?? (NumberFormatException) ??
- ??? return ??? int? ?? long ???? ???? ?? .

??? ??

```

class Solution {
    public long solution(String numbers) {
        String[] words = {
            "zero", "one", "two", "three", "four", "five",
            "six", "seven", "eight", "nine"
        };
        for (int i = 0; i < words.length; i++) {
            numbers = numbers.replaceAll(words[i], String.valueOf(i));
        }
        return Long.parseLong(numbers); // int → long
    }
}

```

??? VS ??????

? ??? (Primitive type)

- int → `int` (primitive int)
- long → `long` (primitive long)

```

int[] arr = {1, 2, 3, 4, 5};
long[] arr2 = {1L, 2L, 3L, 4L, 5L};

```

? ??? (Reference type, Wrapper class)

- Integer → `Integer` (Wrapper class for int)
- Long → `Long` (Wrapper class for long)

```

Integer i = 1;
Long l = 1L;

ArrayList<Integer> list = new ArrayList<>();
list.add(1);
list.add(2);
list.add(3);

```


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[] 배열로 String[] 배열 char[] 배열로 변환 .

? String.toCharArray() ??

- → my_string.toCharArray();
- toCharArray() String 배열로 변환 .
- String.toCharArray() (char[] arr) ()
- String.toCharArray(my_string) char[] 배열로 변환 .

? return arr.toString();

- char[] 배열 toString() String 배열로 변환 .
- → new String(arr) String 배열로 변환 .

?? ??

```

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 ????

static method (static) ** , ,
 method
 .

static method
 .
 Integer.parseInt("123")
 , Integer
 .

, new
 , str.length()
 , str
 .

,
 sayHello()
 ,
 person1.sayHello()
 person2.sayHello()
 .
 name
 .

,
 .
 ,
 .

? 1. ??? (Instance Method)

.

- `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;
    }
}
```

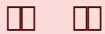
```
}
```

```
// 测试
```

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

5. [Java] ? ?? ??? ?? (???) - sb.toString();

<https://school.programmers.co.kr/learn/courses/30/lessons/120896>



```
import java.util.*;

class Solution {
    public String solution(String s) {
        int[] cnt = new int[26];

        for(char c : s.toCharArray()) {
            cnt[c - 'a']++;
        }

        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < 26; i++) {
            if (cnt[i] == 1) sb.append((char)i+'a');
        }
        return sb.toString();
    }
}
```



기댓값 > "d"

입력값 > "abdc"

입력값 > "hello"

[illegible]

→ i + 'a' char .

```
class Solution {
    public String solution(String s) {
        int[] cnt = new int[26];

        for(char c : s.toCharArray()) {
            cnt[c - 'a']++;
        }
    }
}
```

```
}

StringBuilder sb = new StringBuilder();
for (int i = 0; i < 26; i++) {
    if (cnt[i] == 1) sb.append((char)(i + 'a'));
}
return sb.toString();
}
}
```

10. [Java] ?? ?? - String.indexOf(), String.valueOf()

<https://school.programmers.co.kr/learn/courses/30/lessons/120904>

?? ??

```
class Solution {
    public int solution(int num, int k) {
        String numToString = Integer.toString(num);
        char[] arr = numToString.toCharArray();
        char kToChar = (char) k;
        int idx = -1;
        for(int i = 0; i < arr.length; i++ ) {
            if (arr[i] == kToChar) {
                return i+1;
            }
        }
        return idx;
    }
}
```

? char kToChar = (char) (k + '0');

- ?? ?? : char kToChar = (char) k;
- int ??? char ??? ??? ??
- → int 3 (char) '3' ?? 'ETX' ?? ??
- → char kToChar = (char)(k + '0'); ?? Character.forDigit(k, 10); ?? ??

??? ??

```
class Solution {
    public int solution(int num, int k) {
        String numToString = Integer.toString(num);
        char[] arr = numToString.toCharArray();
```



```
char kToChar = (char) (k + '0');
int idx = -1;
for(int i = 0; i < arr.length; i++ ) {
    if (arr[i] == kToChar) {
        return i+1;
    }
}
return idx;
}
```

□ □□ □□ (String API □□)

```
public int solution(int num, int k) {
    String s = String.valueOf(num);
    int idx = s.indexOf(String.valueOf(k));
    return idx == -1 ? -1 : idx + 1;
}
```

? String.indexOf()

```
String s = "29183";  
int idx = s.indexOf("1");    // 2
```

- `"1"` `"29183"` 0-based index 2
- `-1`

```
int indexOf(int ch)           // 00 00 00 00 (ex: 'a', 97)
int indexOf(String str)      // 0000
```

? ?????

三元运算符 (ternary operator) 是 if-else 的简写形式，用于在单行代码中根据条件选择不同的值。

基本语法

```
条件 ? 如果为真时的值 : 如果为假时的值;
```

- 如果条件为 true，则返回第一个值。
- 如果条件为 false，则返回第二个值。

示例

```
int age = 20;  
String result = (age >= 18) ? "成年" : "未成年";  
System.out.println(result); // 输出: 成年
```