

## String 메서드

메서드/설명	예제	결과
<b>indexOf(str)</b> 인수로 전달받은 문자열을 검색하여 첫 번째 인덱스를 반환한다. 찾지 못하면 -1을 반환한다.	let s = "Hello"; let p1=s.indexOf("e"); let p2=s.indexOf("llo");	p1=1 p2=2
<b>includes(str)</b> 인수로 전달받은 문자열이 포함되어 있는지 확인하여 true, false 를 반환한다.	let s = "Hello"; let p1=s.includes('e'); let p2=s.includes('oo')	p1=true p2=false
<b>substring(begin)</b> begin 인덱스 위치부터 마지막 문자까지의 부분 문자열을 반환합니다.	let s = "Hello World"; let p1=s.substring(6)	p1="World"
<b>substring(begin, end)</b> begin 인덱스 위치부터 end인덱스 바로 앞 문자까지의 부분문자열을 반환합니다.	let s = "Hello World"; let p1=s.substring(1, 8);	p1="ello Wo"
<b>toUpperCase()</b> 문자열의 모든 문자를 대문자로 바꾸어준다.	let s = "Hello World"; console.log(s.toUpperCase());	"HELLO WORLD"
<b>toLowerCase()</b> 문자열의 모든 문자를 소문자로 바꾸어준다.	let s = "Hello World"; console.log(s.toUpperCase());	"hello world"
<b>charCodeAt()</b> 문자의 아스키번호를 반환해준다.	console.log('A'.charCodeAt());	65
<b>String.fromCharCode(아스키번호)</b> 아스키번호의 문자를 반환합니다.	p1=String.fromCharCode(65, 66, 67);	p1="ABC"
<b>replace(s1, s2)</b> 문자열에서 s1문자열을 찾아 s2문자열로 교체한 문자열을 반환한다.	let s = "Hello World"; let p1=s.replace("Hello", "Good");	p1="Good World"
<b>split(str)</b> str에 의하여 문자열을 분리한 후 분리된 문자열을 원소로 갖는 배열을 반환한다.	let s = "It is time to study"; p1=s.split(' ')	p1=['It', 'is', 'time', 'to', 'study']
<b>length</b> 문자열의 길이값이다.	let s = "Hello World"; p1=s.length	p1=11
<b>join(str)-배열의 메서드이지만 알아두자</b> 배열의 모든 요소를 연결해 하나의 문자열을 반환한다. str을 연결문자로 해서 연결한다.	let s = "It is time to study"; p1=s.split(' ') console.log(p1.join('-'));	It-is-time-to-study

## Set

1. Set 객체는 중복되지 않는 유일한 값들을 저장하는 집합입니다.

2. Set 객체 메서드

메서드/설명	예제	결과
<b>const set = new Set();</b> Set 객체를 생성합니다.	<code>const set = new Set([1, 2, 2, 3]);</code>	<code>set={1, 2, 3}</code>
<b>size</b> Set 객체의 요소개수를 확인할 수 있습니다.	<code>const set = new Set([1, 2, 2, 3]);</code> <code>p1=set.size;</code>	<code>p1=3</code>
<b>add(item)</b> Set 객체에 item을 추가합니다.	<code>const set = new Set();</code> <code>set.add(1);</code>	<code>set={1}</code>
<b>has(item)</b> Set 객체에 item 원소가 있는지 확인해서 있으면 true, 없으면 false를 반환한다.	<code>const set = new Set([1, 2, 2, 3]);</code> <code>let p1=set.has(2);</code> <code>let p2=set.has(5);</code>	<code>p1=true</code> <code>p2=false</code>
<b>delete(item)</b> Set 객체에 item 원소를 삭제한다.	<code>const set = new Set([1, 2, 2, 3]);</code> <code>let p1=set.delete(2);</code>	<code>set={1, 3}</code>
<b>clear()</b> Set 객체의 모든 원소를 삭제한다.	<code>const set = new Set([1, 2, 2, 3]);</code> <code>set.clear();</code>	<code>set={}</code>

## 문자 찾기

한 개의 문자열을 입력받고, 특정 문자를 입력받아 해당 특정문자가 입력받은 문자열에 몇 개 존재하는지 알아내는 프로그램을 작성하세요.

매개변수 s에 문자열이 주어지고, 매개변수 c에 찾을 문자가 주어지면 s문자열에 c문자가 몇 개 있는지 확인하여 그 개수를 반환하는 프로그램을 작성하세요.

입출력 예:

s	c	result
"COMPUTERPROGRAMMING"	R	3
"IT IS TIME TO GO"	E	1
"TEACHER"	E	2

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.
- 문자열 s는 대문자로만 이루어져 있습니다.

## 대문자 찾기

한 개의 문자열을 입력받아 해당 문자열에 알파벳 대문자가 몇 개 있는지 알아내는 프로그램을 작성하려 합니다.

매개변수 s에 문자열이 주어지면 s문자열에 대문자가 몇 개 있는지 개수를 반환하는 프로그램을 작성하세요.

입출력 예:

s	result
"KoreaTimeGood"	3
"IT IS TIME TO GO"	12
"Teacher is My friEnd"	3

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

## 대소문자 변환

대문자와 소문자가 같이 존재하는 문자열을 입력받아 대문자는 소문자로 소문자는 대문자로 변환하려고 합니다.

매개변수 s에 대소문자가 존재하는 문자열이 입력되면 문자열 s의 대문자는 소문자로, 소문자는 대문자로 변환하여 반환하는 프로그램을 작성하세요.

입출력 예:

s	result
"StuDy"	"sTUdy"
"TeachER"	"tEACHer"
"Good"	"gOOD"

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.

## 가장 긴 문자열

N개의 문자열이 입력되면 그 중 가장 긴 문자열을 찾으려고 합니다.

매개변수 s에 N개의 문자열이 주어지면 그 중 가장 긴 문자열을 반환하는 프로그램을 작성하세요. 답이 여러개면 s배열에서 제일 먼저 나타나는 문자열을 답으로 합니다.

입출력 예:

s	result
["teacher", "time", "student", "beautiful", "good"]	"beautiful"
["big", "Good", "easy"]	"Good"

제한사항:

- 문자열 배열 s의 길이는 30을 넘지 않습니다.
- 문자열 배열 s의 원소인 문자열은 길이가 100을 넘지 않습니다.

## 중복문자제거

소문자로 된 한개의 문자열이 입력되면 중복된 문자를 제거하려고 합니다.

매개변수 s에 문자열이 입력되면 중복된 문자는 제거하고 모든 문자는 오직 하나만 있게 하여 반환하는 프로그램을 작성하세요.

제거된 문자열의 각 문자는 원래 문자열의 순서를 유지합니다.

입출력 예:

s	result
"ksekkset"	"kset"
"ssssttessskssee"	"stek"

제한사항:

- 문자열의 길이는 100을 넘지 않습니다.

## 공통 문자열

N개의 문자열이 주어지면 이 문자열들의 최대 공통 접두사를 출력하는 프로그램을 작성하세요. 만약 문자열들이 ["long", "longtime", "longest"] 라면 세 단어의 최대 공통 접두사는 "long"입니다.

매개변수 s에 N개의 문자열이 주어지면 각 문자열들의 최대 공통 접두사를 반환하는 프로그램을 작성하세요. 공통 접두사는 반드시 존재합니다.

입출력 예:

s	result
["long", "longtime", "longest"]	"long"
["big", "Good", "easy"]	"Good"

제한사항:

- 문자열 배열 s의 길이는 30을 넘지 않습니다.
- 문자열 배열 s의 원소인 문자열은 길이가 100을 넘지 않습니다.



## 문자열 압축

알파벳 대문자로 이루어진 문자열을 입력받아 같은 문자가 연속으로 반복되는 경우 반복되는 문자 바로 오른쪽에 반복 횟수를 표기하는 방법으로 문자열을 압축하려고 합니다.

매개변수 `s`에 문자열이 입력되면 반복횟수를 표기하는 방법으로 문자열을 압축하여 반환하는 프로그램을 작성하세요. 단 반복횟수가 1인 경우 생략합니다.

입출력 예:

s	result
"KKHSSSSSSSE"	"K2HS7E"
"AAABCCCDD"	"A2BC3D2"

제한사항:

- 문자열 `s`의 길이는 100을 넘지 않습니다.

## 회문 문자열

앞에서 읽을 때나 뒤에서 읽을 때나 같은 문자열을 회문 문자열이라고 합니다.

매개변수 `s`에 문자열이 입력되면 해당 문자열이 회문 문자열이면 "YES", 회문 문자열이 아니면 "NO"를 출력하는 프로그램을 작성하세요.

단 회문을 검사할 때 대소문자를 구분하지 않습니다.

입출력 예:

s	result
"gooG"	"YES"
"Moon"	"NO"

제한사항:

- 문자열 `s`의 길이는 100을 넘지 않습니다.

## 회문문자열2

문자열  $s$ 가 주어지면  $s$ 가 최대 문자 1개까지 지워서 회문문자열이 되면 "YES"를 출력하고, 그렇지 않으면 "NO"를 출력하는 프로그램을 작성하세요.

입출력 예:

s	result
"abcbdcba"	"YES"
"abcabbakcba"	"YES"
"abcacbakcba"	"NO"

제한사항:

- 문자열  $s$ 의 길이는 100을 넘지 않습니다.

입력예제 1 설명 :

abcbdcba에서 4번째 문자인  $b$ 를 제거하면 abcdcba로 회문이 된다. 또는 5번째 문자인  $d$ 를 제거하면 abcbcbba로 회문이 된다.

## 문자열의 숫자화(과제)

영어문자열이 주어지면 숫자로 변환하는 프로그램을 작성하고 싶습니다.

"zero"=0,

"one"=1,

"two"=2,

"three"=3,

"four"=4,

"five"=5,

"six"=6,

"seven"=7,

"eight"=8,

"nine"=9

위와 같이 문자열을 숫자로 변환합니다.

만약 입력된 문자열이 "fivesevenzero"는 570으로 변환됩니다.

만약 입력된 문자열이 "zerofiveseven"는 057이지만 첫 자리 0은 무시하고 57로 변환합니다.

입출력 예:

s	result
"fivesevenzero"	570
"zerofiveseven"	57
"eightsixzerofivesevenfive"	860575

제한사항:

- 문자열 s의 길이는 100을 넘지 않습니다.