

[String]

- `str.toCharArray(): String -> char[]`:
- `String.valueOf(char)` 혹은 `new String(char): char[] -> String`
- `str.charAt(0): String의 문자열 중 0번째 인덱스에 위치하는 문자(char)`.
- `.toUpperCase() / .toLowerCase(): String 문자열 전체를 대문자/소문자로 통일`
- `Character.toUpperCase(c) / Character.toLowerCase(c): 문자 c를 대문자/소문자로 변환`
- `Character.isUpperCase(c) / Character.isLowerCase(c): 문자c가 대문자/소문자인지 확인`
- `Character.isAlpabetic(c): 문자c가 알파벳인지 확인`
- `Character.isDigit(c): 문자 c가 숫자인지 확인`

아스키넘버

→ `Character` 문자 '0' = 48, 문자 '9' = 57. 즉, 문자 '0'~'9'는 48~57. 문자'0' - 48 = 숫자 0.

→ 대문자(65-90), 소문자(97-122): 소문자에서 -32하면 대문자가 된다.

- `String[] senArr = sen.split(" ")`: 문자열을 띄어쓰기 공백(" ")으로 잘라서 단어 단위로 `senArr` 문자 배열에 차례대로 저장.
- `sen.indexOf(" ")`: 문자열 `sen`에서 첫번째 공백(" ")을 찾아 인덱스를 반환. 공백이 없으면 -1 반환.
- `new StringBuilder(str).reverse().toString():` 문자열 `str`을 `StringBuilder`클래스로 바꾼 뒤 `reverse()`메서드를 사용해 뒤집은 후 그 결과를 다시 `toString()`메서드로 `String`으로 전환.
- `new StringBuilder.append(여기에 넣는 값은 어떤 형태든 모두 StringBuilder로 변환되어 추가됨)`
- `sen.replaceAll(regex, replacement):` 대체 메서드. 정규식을 사용할 수 있다.
Ex) `[^A-Z]`: 대문자 A부터 Z가 아닌 것 (꺅쇠 ^: 부정)

```
sen = sen.replaceAll("[^A-Z]", ""); // A-Z가 아니면 지우기
```

- `Integer.parseInt(String sen):` 숫자형의 문자열을 10진수의 정수형으로 반환한다.
- `Integer.parseInt(String sen, int radix):` `String sen`을 `radix` 진수의 숫자로 변환한 후 10진수화 한다 (10진수로 바꾸는 메서드이다). 예를 들면 `Integer.parseInt(sen, 2)` 이라면, `String sen`을 2진수로 변환한 뒤 10진수 정수로 바꾼다 ("1000001" -> 1000001 -> 65).
- `sen.substring(int a, int b):` `String sen`의 a번째 인덱스부터 b-1 인덱스까지 자르기.
- `sen.substring(int a):` `String sen`의 a번째 인덱스부터 끝까지 자르기.
- `sen.replace(char oldChar, char newChar):` `String sen`에 있는 모든 `oldChar`을 `newChar`로 교체.
- `(char) num:` 아스키 넘버 `num`을 문자로 변환한다.

★ `scan.nextInt()` 다음에 `scan.nextLine()` 쓴다면, 중간에 `scan.nextLine()` 한 번 더 써서 개행 문자 지워주기. 혹은 `scan.next() - 스페이스, 공백 전까지 입력받은 문자 리턴 - 쓰기`

`StringTokenizer st = new StringTokenizer(br.readLine(0, " ")); // 공백으로 끊은 (공백 제외) 문자 반환`

- `st.countTokens():` Token의 개수 (단어 개수) 반환

- `st.nextToken()`: 다음 토큰 반환
- `st.hasMoreTokens()`: 남은 토큰이 있는지 확인, 있으면 `true` 반환.

[Arrays]

`Arrays.sort(int[] a)`: a 배열을 오름차순 순으로 정렬

`Arrays.equals(int[]a, int[]a2)`: a배열과 a2배열이 같은지 검사. 같으면 `true` 반환
