

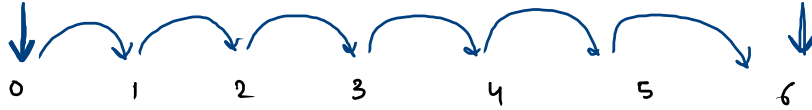
Print Characters

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    String str = scn.nextLine();  
  
    for (int i = 0; i < str.length(); i++) {  
        System.out.println( str.charAt(i) );  
    }  
}
```

⇒ Loop → for each loop

syntax

for (data-type var_name : iterator) {
}



arr

5	7	-2	8	3	1	4
---	---	----	---	---	---	---

for (int i : arr) {
}

value

Note:-

- 1) for each loop can only start from 0th index and will go till last index
- 2) it can only move in forward direction
- 3) it can take only 1 jump at a time
- 4) here, i is value instead of index

code

```
public static void main(String[] args) {  
    int[] arr = { 3, 4, 1, -2, 6, 5, 3 };  
    for (int i : arr) {  
        System.out.println( i );  
    }  
}
```

Is Equal?

str1 = "abcde";

str2 = "abcdef";

Rule :-

- 1) size should be same
- 2) all characters at same index should be same

↓ ↓ ↓ ↓ ↓
str1 = "abcde";
str2 = "abcdf";
↑ ↑ ↑ ↑ ↑

code

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    String str1 = scn.nextLine();  
    String str2 = scn.nextLine();  
    boolean ans = isEqual(str1, str2);  
    System.out.println(ans);  
}  
  
public static boolean isEqual(String str1, String str2) {  
    if ( str1.length() == str2.length() ) {  
        for (int i = 0; i < str1.length(); i++) {  
            if ( str1.charAt(i) != str2.charAt(i) ) {  
                return false;  
            }  
        }  
        return true;  
    } else {  
        return false;  
    }  
}
```

Print Indices of Vowels

str = "Nishukumar" ;
0 1 2 3 4 5 6 7 8 9

vowels = a, e, i, o, u
= A, E, I, O, U

code

```
public static void main(String[] args) {  
    Scanner scn = new Scanner(System.in);  
    String str = scn.nextLine();  
  
    for (int i = 0; i < str.length(); i++) {  
        char ch = str.charAt(i);  
        if ( ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u' || ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U' ) {  
            System.out.print(i + " ");  
        }  
    }  
}
```

Count Words

str = "This _ is _ a _ sentence";

Inbuilt

String[] arr = str.split(" ");

	0	1	2	3
arr =	"This"	"is"	"a"	"sentence"

```

public static void main(String[] args) {
    String str = "abd eaf xyz";
    String[] arr = str.split("a");

    for (int i = 0; i < arr.length; i++) {
        System.out.println( arr[i] );
    }
}

```

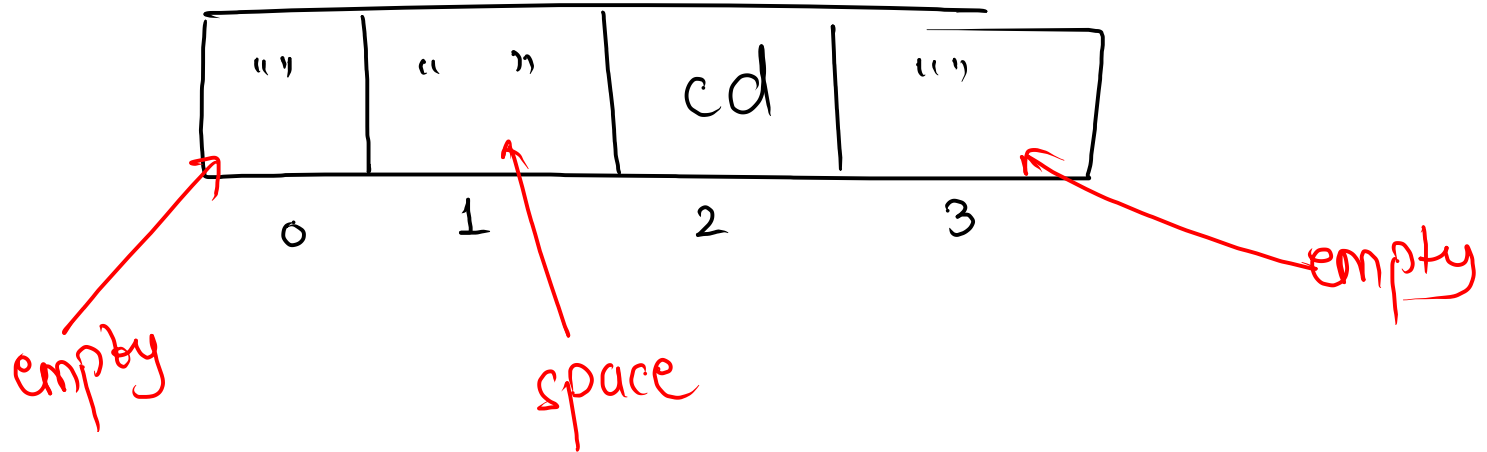
Note:- str = " ab c e ab c d g h ab c ";

str.split("ab"); ← only accept a string

"	"ce"	"cdgh"	"c"
0	1	2	3

str = " ab abcdab "

str.split(" ab ")



Count Words

(cannot use extra memory
& cannot use inbuilt fn)

str = "This is a sentence";

The diagram shows the string "This is a sentence" with a semicolon at the end. Red horizontal lines are placed under each space character. Blue arrows point upwards to every character in the string, including spaces and the semicolon. Three blue curved brackets are drawn below the string to group the words: the first bracket is under "This", the second is under "is", and the third is under "sentence".

curr_ch == ' ' && prev_char != ' '

condition where we can count 1 word is
when we get first space

code

```
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String str = scn.nextLine();

    System.out.println(countWords(str));
}
public static int countWords(String str) {
    int count = 0;
    for (int i = 1; i < str.length(); i++) {
        if ( str.charAt(i) == ' ' && str.charAt(i - 1) != ' ' ) {
            count++;
        }
    }
    if ( str.charAt( str.length() - 1 ) != ' ' ) {
        return count + 1;
    } else {
        return count;
    }
}
```