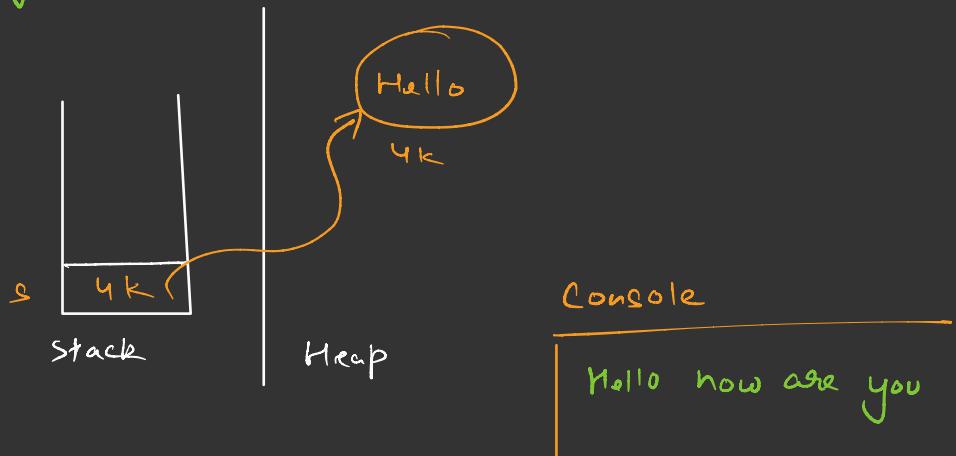


Non Primitive ←  
String ↳ Sequence of characters  
 ↳ " " " 123 abc AE @"  
 ↳ "a" "\n"  
 'Newton School' → Characters only  
 Syntax

String s = "Hello";

Memory



Input

Scanner scn = new Scanner();

String s1 = scn.next(); // Hello

String s2 = scn.next(); // how

String s3 = scn.nextLine() // - are you

String s = "Hey how"  
      <sub>0 1 2 3 4 5 6</sub>

0	1	2	3
a	b	c	d

arr

sysol(arr[2]) // c

char ch = arr[1];

arr[0] = 'd';

charAt(index)

char ch = s.charAt(s);

s.charAt(3) = 'a' X

---

ASCII code

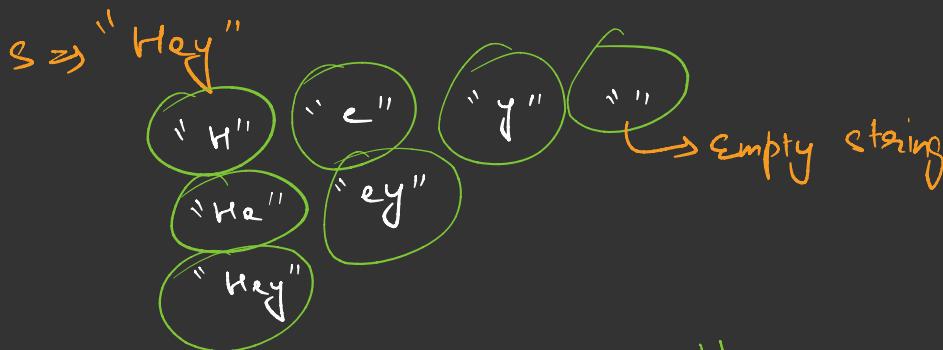
*	0 → 48
\$	9 → 57
a	A 65
=	0
/	2
A	a = 97

char → 2 bytes  
→ 16 bits  
→ 2<sup>16</sup>

char c = 97;  
sysol(c) // a

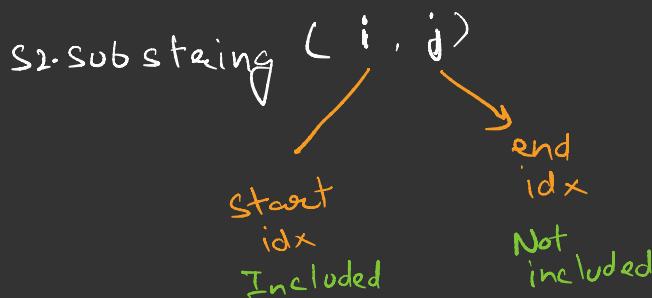
## Substring

Part of a string



11 Length

$s2 = "Hello World"$



$s2.\text{substring}(1, 7);$	// ello w
$(0, 4)$	// Hell
$(7, 8)$	// o
$(4, 10)$	o word
$(9, 11)$	// ld
$(1, 1)$	// "
$(3, 1)$	// Error

$s2.\text{substring}(2,$

## Append Strings (Concatenation)

String a = "Hello";

String b = "World";

String c = a + " " + b;

sysc(c) // Hello World

Convert no into string  
" " + 9857

→ "9857"

c = c + a;

sysc(c) // Hello World9

## Length

s.length()

→ Returns Length of the string

## toCharArray()

char arr[] = c.toCharArray()

0	1	2	3	4	5	6	7	8	9	10	11
H	e	l	l	o		w	o	r	l	d	9

$s = "98475"$

① Integer.parseInt(s)

② "98475"

```
int res = 0;  
for (i=0; i < s.length(); i++) {  
    digit = s.charAt(i);  
    res = res * 10 + (digit - '0');
```

res = 3

9

'9' - '0'

$9 - 0$

$9 - 18$   
 $\rightarrow \textcircled{9}$

0 1 2 3 4 5 6 7 8 9  
48 49 50 51 52 53 54 55 56 57

# Memory Management

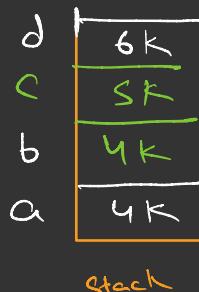
String a = "Hey"

String b = "Hey"

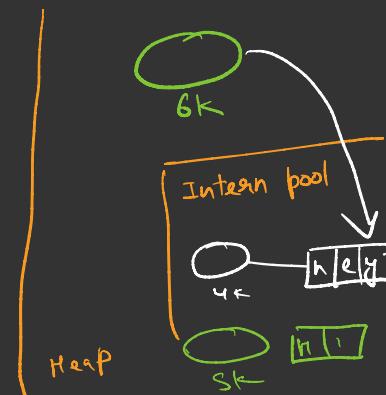
c = Hi

String d = new

String("Hey");



stack



What?

Why → optimise space

Implications

① Comparison

1)  $a == b$  // true

→ checks the address

$a == d$

2) .equals → first compares by address  
compare char by char.

a.equals(b) // true

b.equals(d) // true

## ② Immutability

↳ strings are immutable

↳ reference is mutable  
instance is not

String a = "Hey"

a.setcharAt X

a = "Hi"

a = a + " bye";

## Implications of Immutability

Performance

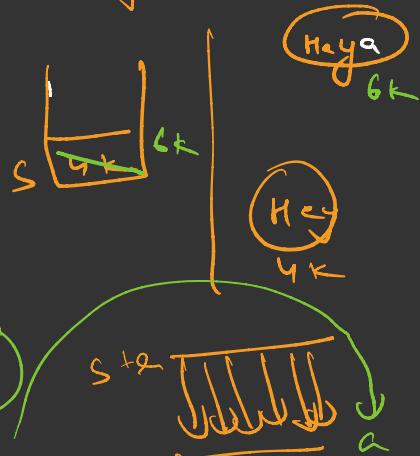
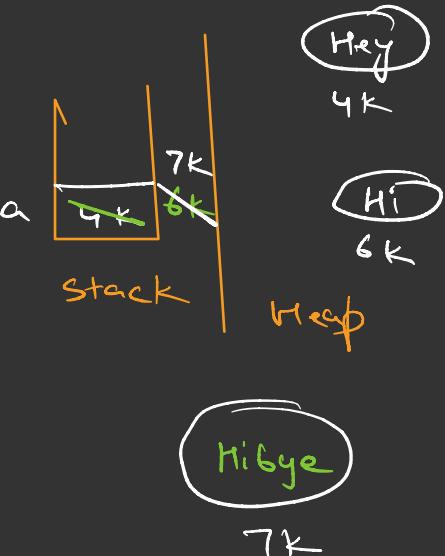
s = "Hey"

s = s + 'a';

Star

2GB

Star = Star + 'a'



Ques Print all the substrings of a string

$S = "Hey"$

$\overset{s \rightarrow}{\underset{\circ}{H}e\underset{\bullet}{y}}$

$H (0, 1)$

$He (0, 2)$

$Hey (0, 3)$

$\overset{\uparrow}{e}$

$e (1, 2)$

$ey (1, 3)$

$y (2, 3)$

s. substring (1 7)

String res = "";

for (int i=1; i<7; i++)

res += s.charAt(i);

return res;

Ques tell if string is Palindrome

mom

ee

madam

c b b a

naman

1 0 2 0 3 0 2 0 1

①  $s = a b c d c b a$

$\equiv rev = a b c d c b a$

$rev = " "$

`for(i=0; i < s.length(); i++)`

$rev = s.charAt(i) + rev;$

Time  $\Rightarrow O(n)$

Space  $\Rightarrow O(n)$

②  $a b c c b a$

$\uparrow \quad \uparrow$   
 $l \quad r$

$s.charAt(l) == s.charAt(r)$

Ques Make string uppercase

$s = "aBccedD" \rightarrow "ABCcedD"$

$\begin{array}{r} a \quad b \\ \hline A \quad B \end{array}$

$\begin{array}{r} a \quad e \\ \hline H \quad E \end{array}$

$ch - 'a' + 'A'$

$a - 97$

e  
E

$A - 65$   
 $32$

small case -  $32 \Rightarrow$  uppercase

small case -  $('a' - 'A') \Rightarrow$  uppercase