

Value you
pass in string is the literal

Date _____
Page _____

#String is basically a set of characters
& it is enclosed in double quotes

String str = "Hello world"

→ string literal

In memory it is stored like

→

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

str.

str hold address of ~~area~~ string

In Java ~~as~~ you can convert array of character into string like

char c[] = { 'H', 'E', 'L', 'L', 'O' }
↑

Character should always be in single quotes

String is

predefined
class in
Java

→ String str = new String(c);

now str = "HELL";

String is
acts like
constructor

that's how we can convert array of characters to string

⇒ You also can use byte make a string

Ex: → as we know characters are made up from ~~any~~ a byte value so we can make a string from byte values also

Note
~A = 65 a = 97
Z = 91 z = 122

Date _____
Page _____

Ex: ⇒

Byte arr = { 65, 66, 67, 68 }
A B C D

so By using constructor convert byte array into a string

String arr = new String(arr);

We are using byte to make int

arr = "ABCD";

arr → | A | B | C | D |

If you pass value which is out of limit of byte will fetch error

⇒ When you call constructor object will be created inside the heap

Also say for some reason you want to create a string but not in program stack but in heap memory

so do like below

⇒ String str = new String("ABCD");

#

str ⇒ | A | B | C | D |

Ex: ⇒ String d = "ABCD";

String str = new String(d);

pool is part of Java memory

Date _____
Page _____



As you use constant values directly in code & you're not pointing it with any variable so it can be said that value is present in pool.

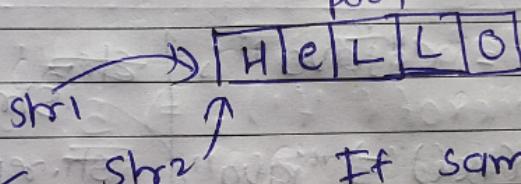
You only create value in heap if you use word here

EX

String str1 = "Hello";

String str2 = "Hello";

pool



If same value is there two or more variable can point at the same value (Java intelligent)

This because

As you created object normally so it will created inside the pool.

Pool is

Pool is a area in heap memory where java literals are being stored

so If we want specific character from a character array then

`Char c[] = { 'A', 'B', 'C', 'D' };`

If you only want C & D

`String cc = new String(c, 2, 2);`

`String(characters, startingPoint, length);`
so it will only point at
`cc = [C]D`

It is simple & well mannered way
for slicing also

`String str = "Java";`
methods,

`str.toLowerCase() => "java"`

`str.toUpperCase() => "JAVA"`

`String str = " welcome "`

`str.trim() => "welcome"`

`str.substring(int begin, int end)`

`str.substring(2) => "come";`

`str.substring(2, 4) => "co"`

`str.substring(0, 4)`

`=> Welcome`

`str = "Java"`
`str.replace('a', 'o')`

`str => "JOVO"`

`str.startsWith("Ja")`
`=> true`

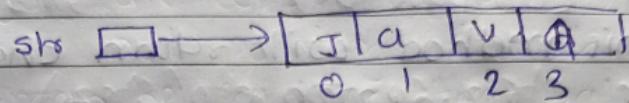
`str.endsWith("va")`
`=> true`

output strings after method manipulation are get created in heap.

#String methods

In all method element of old are not manipulated we will get entirely new string.

String str = "JAVA"; \Rightarrow created in pool



1] str.toLowerCase() \Rightarrow convert all character to lower case alphabet

2] str.toUpperCase() \Rightarrow convert all characters to upper case alphabets

3] str.trim() \Rightarrow It will remove blank spaces in a string, get

4] str.substring(2) \Rightarrow It will print all elements after 2 (including position 2)

5] str.substring(2, 4) \Rightarrow It will print element from 2 to one element less than 4 is exclusive
(start index \Rightarrow Inclusive)
(End index \Rightarrow Exclusive)

substring is subset of original string

\Rightarrow str.replace(char old, char new)

it will replace old character with new one

④ boolean startWith(String s)
boolean endWith(String s)

Both function with check if string is starting or ending with string you specify. It will return output in boolean true or false format.

Ex: \Rightarrow check a string is welcome or not it is not

.org \Rightarrow organization
.net \Rightarrow network

.gov \Rightarrow government
.com \Rightarrow commercial

You can pass any string as parameter to this method.

(*) str.charAt(@number)

This method will return character at that index

String Str = "abc^{0 1 2 3}def";

Str.charAt(3); \Rightarrow "d"

In you can send starting point

\Rightarrow Str.indexOf("character or string");

this will return index of string or character

return

-1 if invalid

str.indexOf('f'); \Rightarrow 5

Str.indexOf('ef')

first character

If you want to search from index

str.indexOf('f', 3); \Rightarrow 5

Search start after

first string will be fetch & then index of first element e here if it is i.e. so index of e is \Rightarrow returned.

(*) str.lastIndexOf()

It is same as indexOf but index of search from start & last index will search from last to first

String Str = "ABCDEFGHI";

Search from last
→

Date _____
Page _____

str.lastIndexof('f'); ⇒ 5

Here also you can fetch the starting
Index of →

str = "A B C D E F G H I" ⇒ starting po.
0 1 2 3 4 5 6 7 8 ⇒ lastIndexof

str.lastIndexof('E', 7) (left R → L)

str.indexOf('E', 2) (L → R)

Index of search for element in right direction
of starting point lastIndex of search from
left direction of starting point

Equals

str1 = "Hello" str2 = "Java"

str3 = " HELLO", str4 = " Java";

+ boolean

Eq ⇒ str.equals() ⇒ It will return output
as true or false & check that is strings
are exactly equal or not

str2.equals(str4) ⇒ true

str1.equals(str3) ⇒ false

(case are not same)

So If you want to check strings are
equal or not & you don't have case condition

str2.equalsIgnoreCase(case(str2)) ⇒ true

str1.equalsIgnoreCase(case(str3)) ⇒ true

→ so now case will ignored & only letters are checked

⊕ compareTo()

str2.compareTo(str3)

It is dichotomous function check which letter comes first

If str2 letter is first -1
(first string is smaller than 2)

+ str3.compareTo(str2)

P > J ⇒ +1

If first string > second string ⇒ +1
Second string > first string ⇒ -1

Upper Case are smaller than lower

⇒ It basically checks ASCII of first letter & one which has higher ASCII code is bigger

⇒ Equals method

str1.equals(str2) → Created in heap

If str1.equals(str2)

and if (P=0R P1=0) str1.equals(str2)

and if (P=1R P1=1) str1.equals(str2)

Create in P00 output ⇒ true

Str1 = \$ = Str2 \Rightarrow false
pool reference not match

In equal we not check for reference we
check character by character

#

String.valueof()

we can convert any type to String

Ex str Int a = 101213

str C = String.valueof(a)

C = "101213"

\Rightarrow we can convert any datatype to string
with help of String.valueof function.