String In Java (Part I)

Strings which are widely used in Java programming, are a sequence of characters. In Java programming language, strings are treated as objects. The Java platform provides the String class to create and manipulate strings.

Note that a Char is a single alphabet whereas String is zero or a sequence of characters. char is a primitive type whereas a String is a class.

char like ‘a’

String like “Iraq”

Creating Strings

1- Literal: The most direct way to create a string is to write:

```
String greeting = "Hello world!";
```

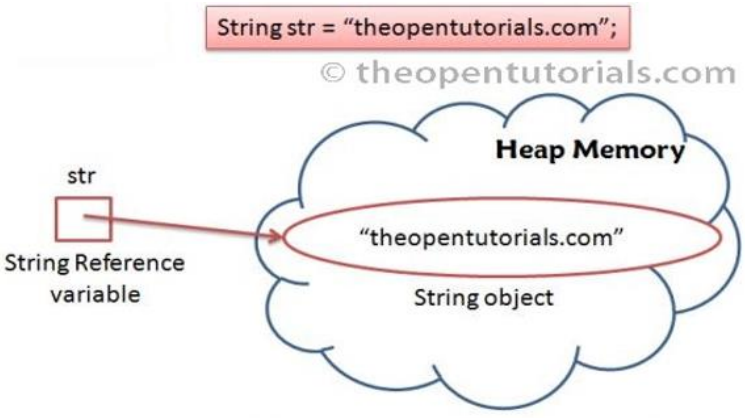
In this case, "Hello world!" is a string literal—a series of characters in your code that is enclosed in double quotes. Whenever it encounters a string literal in your code, the compiler creates a String object with its value—in this case, Hello world!

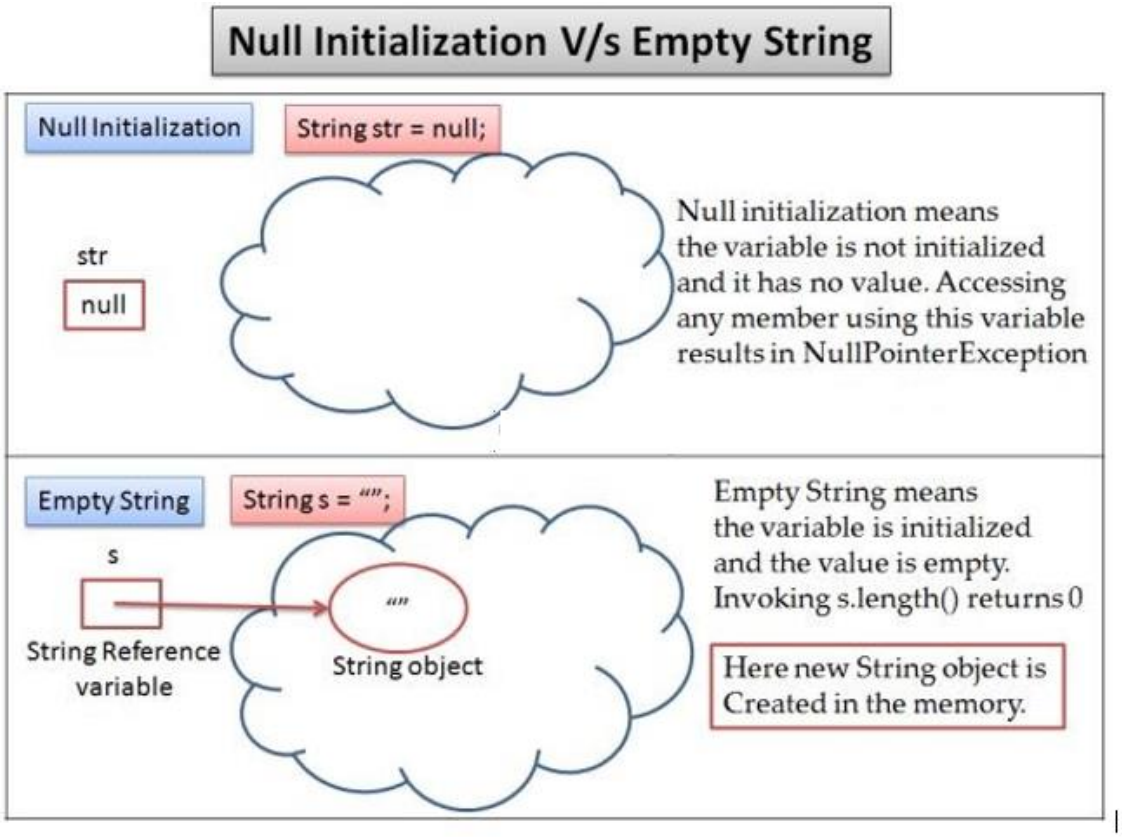
2- Constructor with array of char: As with any other object, you can create String objects by using the new keyword and a constructor. The String class has thirteen constructors that allow you to provide the initial value of the string using different sources, such as an array of characters:

```
char [] a = { 'h', 'e', 'l', 'l', 'o', '.' };
String s = new String(a);
```

3- Constructor with Literal the string could be created using the new keyword and a constructor.

String s=new String(“Welcome”);





Example 1:

Trace the following java code:

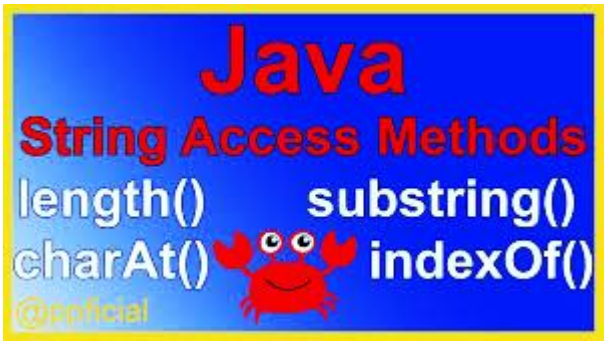
```
public class Example1 {
public static void main(String args[]){
String s1="java";
char ch[]={'s','t','r','i','n','g','s'};
String s2=new String(ch);
String s3=new String("example"); }}
```

Example 2:

Trace the following java code:

```
public class JavaApplication52 {
   public static void main(String[] args) {
      String s1;
      String s2=null;
      String s3=new String();
      String s4="";
      System.out.println(s1);          System.out.println(s2);
      System.out.println(s3);         System.out.println(s4);    } }
```

Java String Methods



- **Java String length()**: The Java String length() method tells the length of the string. It returns count of total number of characters present in the String. For example:

```
public class Example{
public static void main(String args[]{
String s1="hello";
String s2="whatsup";
System.out.println("string length is: "+s1.length());
System.out.println("string length is: "+s2.length());
}}
```

Here, String length() function will return the **length 5 for s1 and 7 for s2** respectively.

- **Java String concat() :** The Java String concat() method combines a specific string at the end of another string and ultimately returns a combined string. It is like appending another string. For example:

```
public class ConcatExample{
public static void main(String args[]){
String s1="hello";
s1=s1.concat("how are you");
System.out.println(s1); }}
```

The above code returns “hellohow are you”.

- Java String toLowerCase() : The java string toLowerCase() method converts all the characters of the String to lower case. For example:

```
public class StringLowerExample{
public static void main(String args[]){
String s1="HELLO HOW Are You?";
String lower=s1.toLowerCase();
System.out.println(lower);}}
```

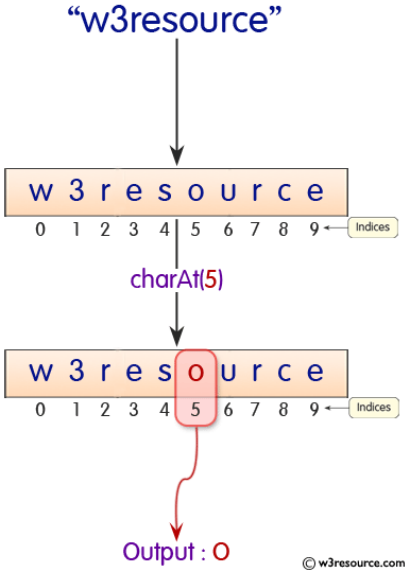
The above code will return "hello how are you".

Java String toUpper() : The Java String toUpperCase() method converts all the characters of the String to upper case. For example:

```
public class StringUpperExample{
public static void main(String args[]){
String s1="hello how are you";
String upper=s1.toUpperCase();
System.out.println(upper); }}
```

The above code will return "HELLO HOW ARE YOU".

- The **charAt()** method of the String class returns the char value at the specified index. An index ranges from 0 to length() - 1. The first char value of the sequence is at index 0, the next at index 1, and so on, as for array indexing.



Example

String s=”IRAQ”; char c=s.charAt(2);

System.out.println(c);

The output is A

Example

String s=”IRAQ” ; char c=s.charAt(4); System.out.println(c);

ERROR …Why?

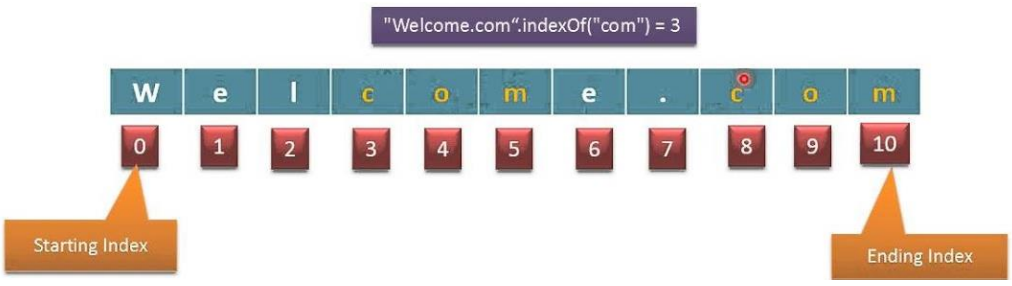
- The method **indexOf()** is used for finding out the index of the specified character or substring in a particular String. There are 4 variations of this method:

  1- int indexOf(int ch): It returns the index of the first occurrence of character ch in a String.

  2- int indexOf(int ch, int fromIndex): It returns the index of first occurrence if character ch, starting from the specified index “fromIndex”.

  3- int indexOf(String str): Returns the index of string str in a particular String.

Method	Description
int indexOf(String str)	Returns the index of the first occurrence of the specified substring.



4- int indexOf(String str, int fromIndex): Returns the index of string str, starting from the specified index “fromIndex”.

All the above variations **returns -1** if the specified char/substring is not found in the particular String.

Example:

```
public class CharAt_IndexOf_Example {
   public static void main(String args[]){
      String str = "This is tutorialspoint";
         System.out.println("Letter at the index 8 is   "+str.charAt(8));
    System.out.println("Index of letter 't' = "+ str.indexOf('t');
    System.out.println("Index of letter 't' = "+ str.indexOf('t',14);
} }
```

Output:

Index of letter 't' = ----- ?

Letter at the index 8 is   ------?

Index of letter 't' = ------ ?

- **Substring in Java**

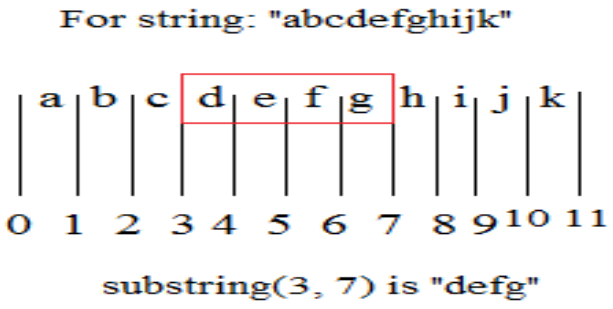
Substring is a subset of another string. Note: Index starts from 0.

You can get substring from the given string object by one of the two methods:

1- public String substring(int startIndex): This method returns new String object containing the substring of the given string from specified startIndex (inclusive).

2- public String substring(int startIndex, int endIndex): This method returns new String object containing the substring of the given string from specified startIndex to endIndex.

In case of substring startIndex is inclusive and endIndex is exclusive.



Example :

Let's understand the startIndex and endIndex by the code given below.

```
String s="hello";
System.out.println(s.substring(0,2));
System.out.println(s.substring(2));
```

The output is he
The output is llo