String Handling in JAVA :-

- String, which are widely used in JAVA Programming, are a Sequence of Characters.
- IN JAVA Programming Language, Strings are treated as Objects.

* String as a Object, that represents Sequence of Char value.

- An array of Characters works Same as JAVA String.

for example :-

```
Char ch = {'j', 'p', 'w', 'e', 'b'};
String s = new String (ch);
```

is same as :-

```
String s = "jpweb";
```

Note :- 1. String are immutable in JAVA.
2. It Contains methods to Perform Operations on Strings such as Compare(), Concat(), equals(), length(), replace() etc...

[String is represented by String class which is located into java.lang Package.

__/__/____

Creating a String Object :-

There are two ways to create String Object :-

1. By String literal
2. By new keyword

1. String literal :-

→ Java String literal is created by using double quotes.
→ String literal is a simple String enclosed in double quotes. " "
→ A String literal is treated as a String Object.

Example :-

```
Public Class Demo
{
Public Static void main (String [] args)
{
String S = "JPweb developers";
System.out.Println(S);
}
}
```

2. Using new keyword :

We can create a new String Object by using new operator that allocate memory for Object.

```
Public class Demo
{
Public static void main (String [] args)
{
String S= new String ("JPwebdevelopers");
System. out Print/n (S);
}
}
```

An example that shows how to declare String :

```
import java.lang.*;
Class Demo
{
Public static void main (String [] args)
{
String S = "JPwebdevelopers";
System.out.Print/n ("String S= "+S);
String S1 = new String ("JP webdevelopers")
System.Out.Println ("String S1 = "+S1);
}
}
```

__/__/____

Output

String s = "JPwebdevelopers";
String s1 = JPwebdevelopers;

* String Operations

Java String class provides various methods to perform different operations on string.

Commonly used String Operations :-

1. length() :-

To find the length of a string, we use the length() method of the String.

Example

```
class Main
{
Public static void main (String[] args)
{
String s1 = "JPnotes";
int length = s1.length();
System.out.println (length);
}
}
```

Output : 7

__/__/____

(2) String Concatenation :-

It is used to combine the two strings. There are two ways to concatenate strings in java.

(i) By + Operator

Java String Concatenation Operator (+) is used to add strings.

```
class abc
{
  Public static void main (String args[])
  {
    String s = "Jp" + "Notes";
    System.out.println (s);
  }
}
```

Output :- JpNotes

(2) By concat() Method

you can also use the concat() method to concatenate two strings :

```
Class abc
{
```

__/__/____

```
public static void main (String args[])
{
    String s1 = "Jp";
    String s2 = "Notes";
    String s3 = s1.concat(s2);
    System.out.println(s3);
}
}
```

Output :- Jp Notes

(3) Compare Strings :-

In Java we can make comparisons between two strings using the equals() method.

```
class abc
{
public static void main(String[] args)
{
    String first = "Java";
    String second = "Java";
    String third = "php";

    System.out.println(first.equals(second));
    System.out.println(first.equals(third));
}
}
```

Output :- true
false

__/__/____

* Substring :=

→ A part of String is called substring.

→ The substring() method extracts a substring from the string and returns it.

Syntax :- (i) String.substring(int startIndex);
↑ beginning index

(ii) String.substring(int startIndex, int endIndex);
↑ beginning ↑ ending

Example :-

```
class Main
{
  public static void main(String[] args)
  {
    String s1 = "jpnotes";
    System.out.println(s1.substring(0,4));
  }
}
```

Output :- jpnot

__/__/____

* StringBuffer class

- StringBuffer class is used to create a mutable string object.
  - ↳ It means, it can be changed after is created.

- So StringBuffer class is used when we have to make a lot of modifications to our string.

Example

```
class Main
{
public static void main (String args[])
{
StringBuffer sb = new StringBuffer ("jpnotes");
System.out.println (sb);

sb.append ("jpwebdevelopers");
System.out.println (sb);

}
}
```

__/__/____

* Difference between String and StringBuffer

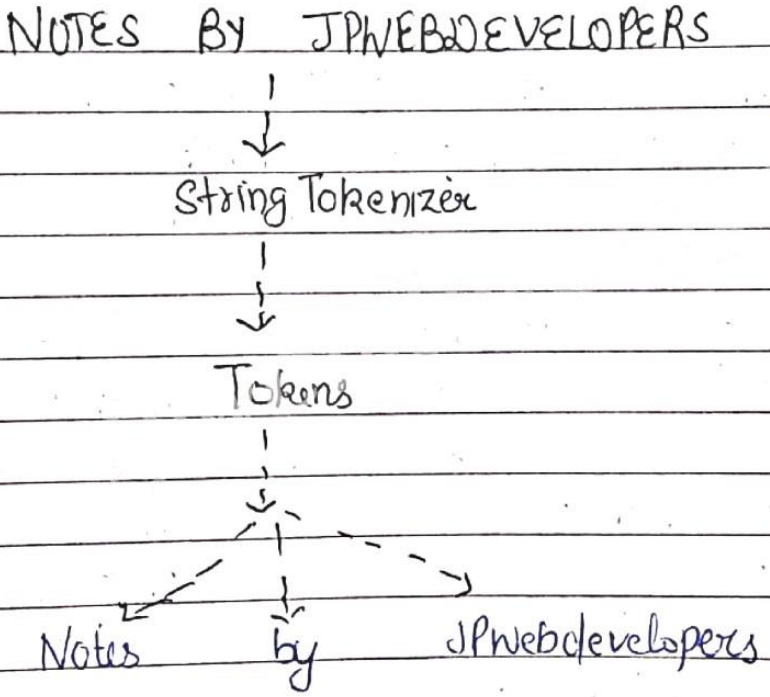
String	StringBuffer
String class is Immutable.	The StringBuffer class is mutable.
The length of the String object is fixed.	The length of the StringBuffer object can be increased.
It is slower during concatenation.	It is faster during concatenation.
Consumes more memory.	Consumes less Memory.
With the String class, you cannot create modifiable string.	The StringBuffer class is used to create modifiable string.
equals method is overridden.	equals method is not overridden.

__/__/____

* String Tokenizer in Java

It is simple way to break a string.

The java.util.StringTokenizer class allows you to break a string into tokens



Methods

(a) countTokens() Returns the total number of tokens present

(b) hasMoreTokens() It Tests if tokens are present for the StringTokenizer's string.

(c) nextToken() It Returns the next token from the given StringTokenizer.

(d) nextElement() It Returns an object rather than String.

(e) hasMoreElements() It Returns the same value as hasMoreToken.

Example

```
import java.util.*;
class Demo
{
public static void main (String args[])
{
StringTokenizer st = new StringTokenizer(" Notes by Prabhdeep");
while (st.hasMoreTokens())
    {
    System.out.println(st.nextToken());
    }
}
}
```