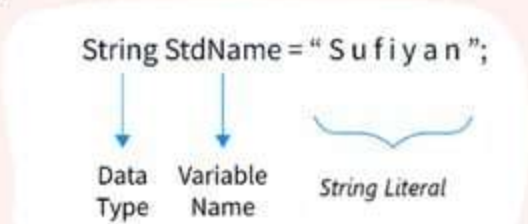
WHAT IS STRING?

- String is non-primitive data type and it is also class which is under java.lang package
- String is a collection of Characters.
- String is immutable.
- it introduces in JDK1.1

HOW TO CREATE A STRING OBJECT?

String object can be created using two ways:

- String Literal.
- new keyword.



1) By String Literal.

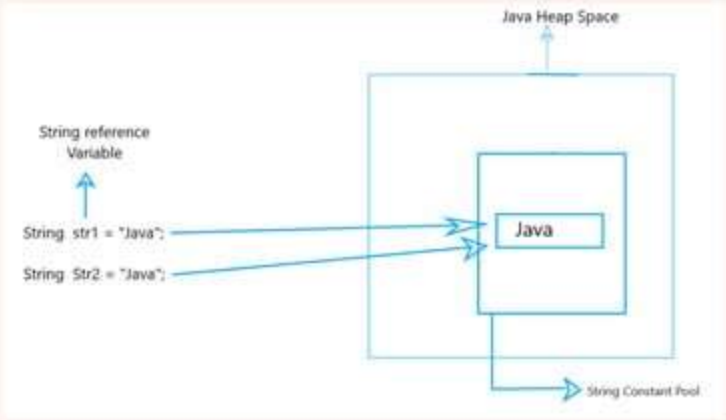
Java String literal can be created and represented using the double-quotes.

All of the content/characters can be added in between the double quotes.

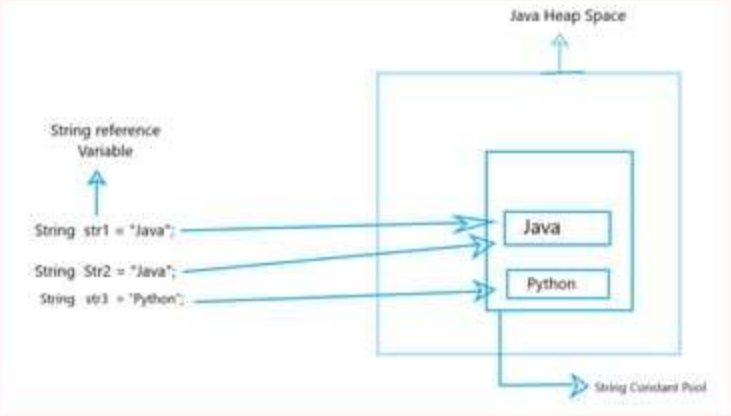
The string literal is always created in the string constant pool.

In Java, String constant pool is a special area that is used for storing string objects.

```
public class Main {
  public static void main(String args[]) {
   String Str1 = "Java";
   String Str2   = "Java";
  }
}
```



```
public class Main {
 public static void main(String args[]) {
  String Str1 = "Java";
  String Str2   = "Java";
   String str3   = "Python";
 }
}
```



2) By New Keyword.

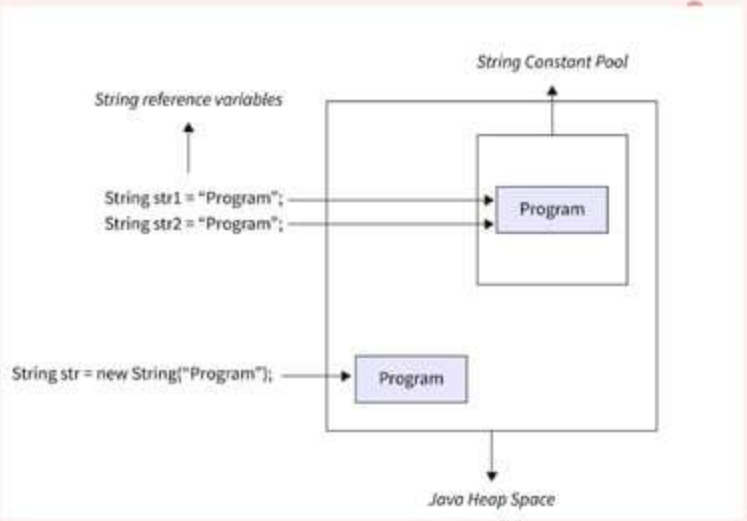
string is created with new, a new object of the String class is created in the heap memory, outside the string constant pool.

Syntax:

```
String stringName = new String("string_value");
```

Example:

```
String str = new String("Program");
System.out.println(str); //Program
```


String Constant Pool
String reference variables
String str1 = "Program";
String str2 = "Program";
Program
String str = new String("Program");
Program
Java Heap Space

EXAMPLE 1

```
public class Main{

 public static void main(String[] args) {
  String str1 = "java";
  String str2 = "programming";
  System.out.println(str1.length()); // 4

     String str4 = str1.concat(str2);
     System.out.println(str4);  // javaprogramming

    }
}
```

== OPERATOR VS EQUALS()

```
String str1 = "java";
 String str2 = new String("java");


   if(str1 == str2){
     System.out.println("both string are equal");
   } else{
      System.out.println("both string are not equal");
   }

   if(str1.equals(str2)){
     System.out.println("both string are equal");
   } else{
      System.out.println("both string are not equal");
   }
```

EQUALS() VS EQUALIGNORECASE()

```
String str1 = "java";
 String str2 = new String("Java");


   if(str1.equals(str2)){
     System.out.println("both string are equal");
   } else{
      System.out.println("both string are not equal");
   }


   if(str1.equalsIgnoreCase(str2)){
     System.out.println("both string are equal");
   } else{
      System.out.println("both string are not equal");
   }
```

WHAT IS STRING BUFFER?

- StringBuffer is mutable String
- StringBuffer Class is (synchronized) thread safe
- it is safe and will result in an order

```
StringBuffer stringBuffer = new StringBuffer("Hello");
stringBuffer.append(" World");
```

EXAMPLE

```
class Main{
public static void main(String args[]){
StringBuffer sb=new StringBuffer("Hello ");
StringBuffer sb1 = new StringBuffer("Hi Java");
sb.insert(1,"Java");//now original string is changed
System.out.println(sb);//prints HJavaello

sb.delete(1,3);
System.out.println(sb);//prints Hvaello

sb1.reverse();
System.out.println(sb1);//prints avaJ iH

sb1.replace(0,4,"Java");
System.out.println(sb1);//prints Java iH
}
}
```

WHAT IS STRING BUILDER?

- StringBuilder is mutable String
- The java StringBuilder class is same as StringBuffer class except that it is non-synchronized (not-thread-safe)
- it is available since JDK 1.5

```
StringBuilder stringBuilder = new StringBuilder("Hello");
stringBuilder.append(" World");
```

STRING BUFFER VS STRING BUILDER

StringBuffer Class	StringBuilder Class
StringBuffer is present in Java.	StringBuilder was introduced in Java 5.
StringBuffer is synchronized.	StringBuilder is asynchronized.
Due to synchronization, StringBuffer is called a thread safe class.	Due to its asynchronous nature, StringBuilder is not a thread safe class.
Due to synchronization, StringBuffer is lot slower than StringBuilder.	Since there is no preliminary check for multiple threads, StringBuilder is a lot faster than StringBuffer.

THANK YOU