String in Java-

String is the class that represents sequence of character.

Package is Java. Lang.

String class implements Serializable, comparable, char sequence interface.

String is the immutable, once string object is created, it cannot changed but new string object is created.

For mutable class, you can use String buffer and String builder class.

How to create the string object?

1. String literal
2. By new keyword

1. String literal-

It is created by using double quotes.

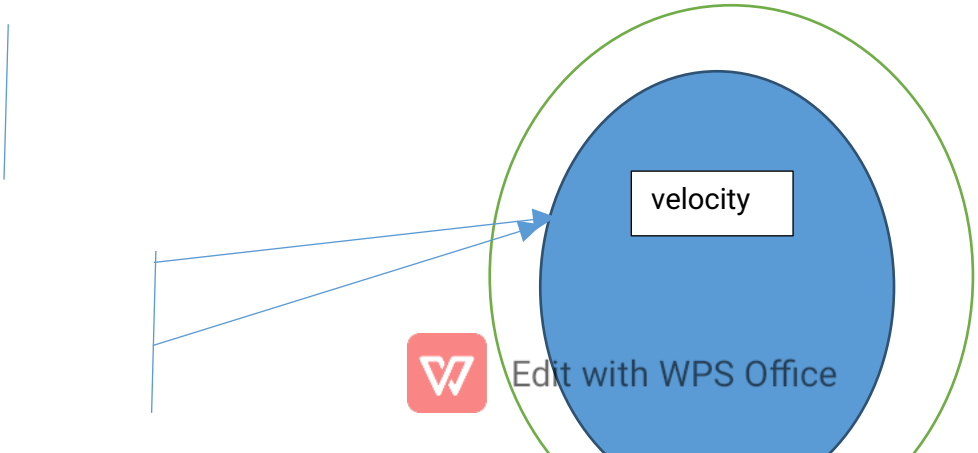
Example- String s=" velocity";

Each time when you create string literal, the JVM check string constant pool first, if the string object is already present in the pool, reference to pooled instance is returned. If string does not present in the pool, new string instance is created and placed in pool.

Example- String s1="velocity";

String s2=" velocity"; //will not create the new instance.

In the above example, only one object will be created, firstly JVM will not find any string object with value "velocity" in the string constant pool, so it will create new object. After that it will find string with value= "velocity" in pool, it will not create the new object but will return reference to same instance.



Why Java uses the concept of string literal?

To make the java more memory efficient (because no new object is created if it exist already in string constant pool.)

2. By new Keyword-

Example- String s= new String("pune");

/* create two objects */

In such case, JVM will create the new String object in normal(non-pool) heap memory and literal "pune" will be placed in string constant pool. The variable s refer to object in heap(non-pool).

How many Objects will be created??

```
String s1 = new String("velocity"); // Two objects created. One  in heap and one in SCP. S1 points to heap object
String s2 = "velocity"; // No object created. s2 point to SCP velocity
String s3 = new String("training"); //Two Objects created. One in heap and one in SCP. s3 point to Heap
String s5 = new String("velocity"); // One Object created in Heap
String s6 = "training"; // No object created. s6 point to SCP training
```

Total : 5 objects will be created

Note:

Whenever we are using new operator compulsory a new object will be created on the Heap . There may be a chance of existing two objects with same content on the heap but there is no chance of existing two objects with same content on SCP . i.e., duplicate objects possible in the heap but not in SCP

For every String Constant one object will be created in SCP. Because of runtime operation if an object is required to create compulsory that object should be placed on the heap but not SCP.

Example- 1

```
package com.velocity;

public class StringDemo {

	public static void main(String[] args) {

		String s1 = "velocity";
		String s2 = new String("velocity");
		System.out.println(s1 == s2);   //false
		System.out.println(s1.equals(s2));  //true

	}
}
```

Example- 2

```
package com.velocity;


public class StringDemo {

	public static void main(String[] args) {

		String s1 = "velocity";
		String s2 = new String("pune");
		s2=s1;
		System.out.println(s1 == s2);  //true
		System.out.println(s1.equals(s2)); //true

	}
}
```

Output-

Example-3

```
package com.velocity;


public class StringDemo {

	public static void main(String[] args) {

		String s1 = new String("velocity");
		String s2 = new String("pune");
		s2 = s1;

		System.out.println(s1 == s2);  //true
		System.out.println(s1.equals(s2));  //true

	}
}
```

Output-

Example- 4

```
package com.velocity;

public class StringDemo {

	public static void main(String[] args) {


public class PropertiesFileDemo {

	public static void main(String[] args) {

		String s1 = "velocity";
		String s2 = new String("pune");
		System.out.println(s1 == s2);  //false
```

```
        System.out.println(s1.equals(s2)); //false
        System.out.println(s1.hashCode()); //100
        System.out.println(s2.hashCode());  //101

    }
}
```

Output-

Example-5

```
package com.velocity;

public class StringDemo {

    public static void main(String[] args) {

        String s1 = new String("velocity");
        String s2 = new String("pune");
        System.out.println(s1 == s2);  //false
        System.out.println(s1.equals(s2));  //false

    }
}
```

Output-

Interning of String Objects :

By using heap object reference, if we want to get corresponding SCP object , then we should go for intern() method.

```
    public static void main(String[] args) {
        String s1 = new String("velocity");
        String s2 = s1.intern();
        System.out.println(s1 == s2); // false
        String s3 = "velocity";
        System.out.println(s2 == s3);// true
    }
```