

Topic: the String class  
(continued)

String s;

s = "" + 3; // "3"

s = 3 + ""; // "3"

Java can convert numbers  
to a String representation  
of the number.

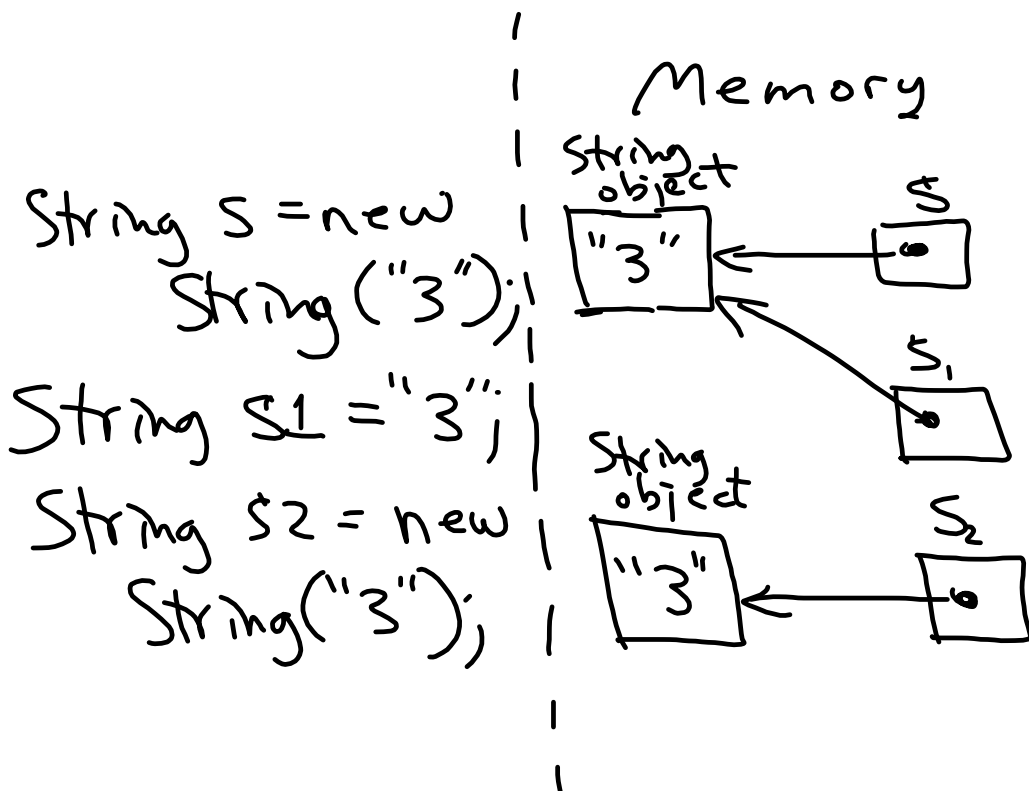
String objects are  
immutable. No method  
of the String class  
changes the String  
object. Usually they  
return a new String  
object.

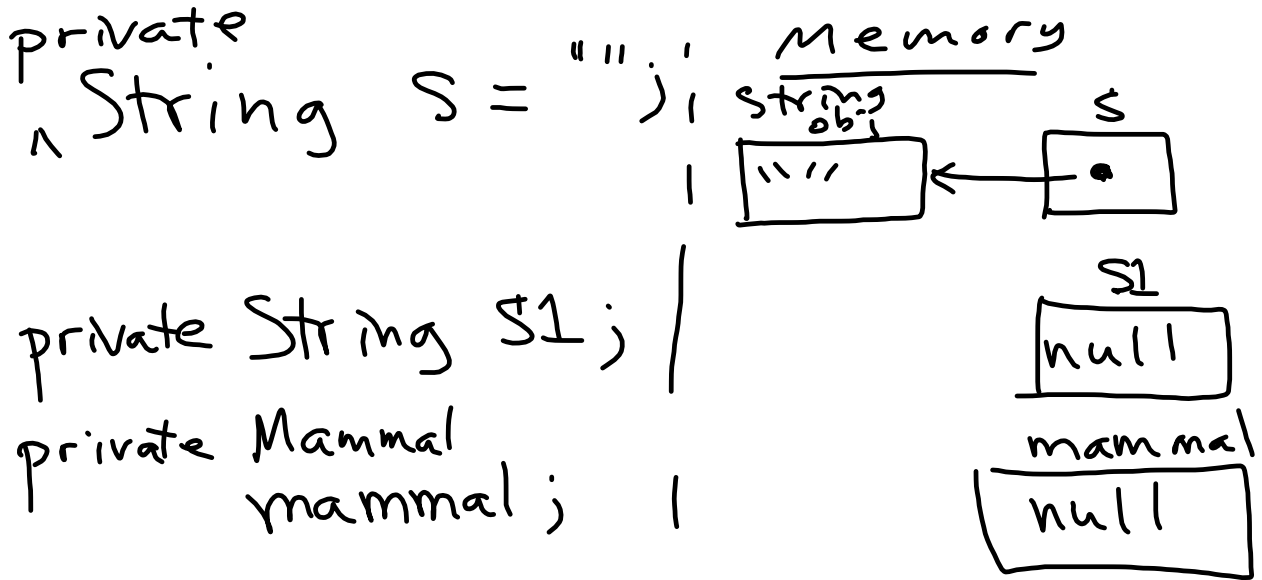
String s = "3";

- DLS of String class -  
you don't need "new"  
to construct a String  
object.

String s = new String("3");

- use of "new" guarantees  
that a new object will  
be created.





mammal == null // true

```
String s;
```

Note: Codepad variables are automatically initialized in the same way as instance fields.

```
s = 3;
```

Error: incompatible types

```
s = "" + 3;
```

```
s
```

```
"3" (String)
```

```
s = 3 + "";
```

```
s
```

```
"3" (String)
```

```
s = (String)3;
```

Error: inconvertible types

required: java.lang.String

found: int

```
s = "3";
```

```
s
```

```
"3" (String)
```

Topic the compareTo method

s1.compareTo(String s2)

expression returning int  
returns

s1 < s2	< 0
s1 equals s2	0
s1 > s2	> 0

```
String s1 = "Aardvark";
```

```
String s2 = "Bear";
```

```
s1.compareTo(s2)
```

```
-1 (int)
```

```
s1 = "aardvark"
```

```
"aardvark" (String)
```

```
s1.compareTo(s2)
```

```
31 (int)
```