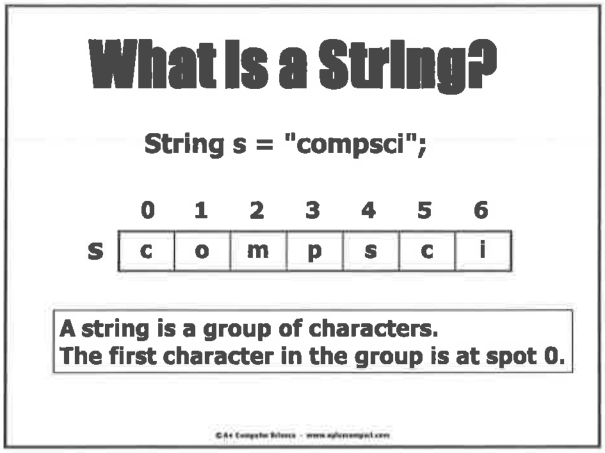


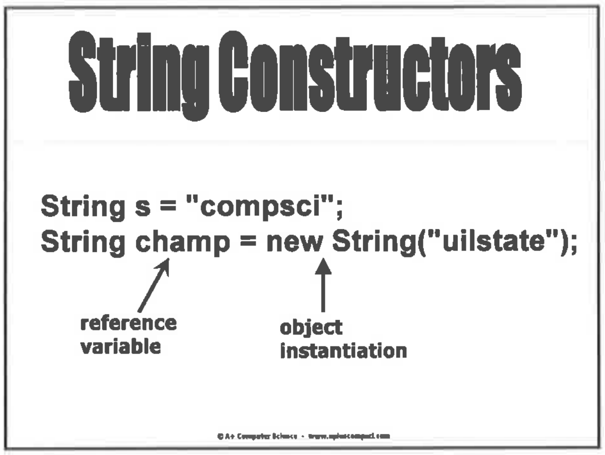
What Is a String?

String s = "compsci";

	0	1	2	3	4	5	6
s	c	o	m	p	s	c	i

A string is a group of characters.
The first character in the group is at spot 0.

©A+ Computer Science - www.apluscompsci.com

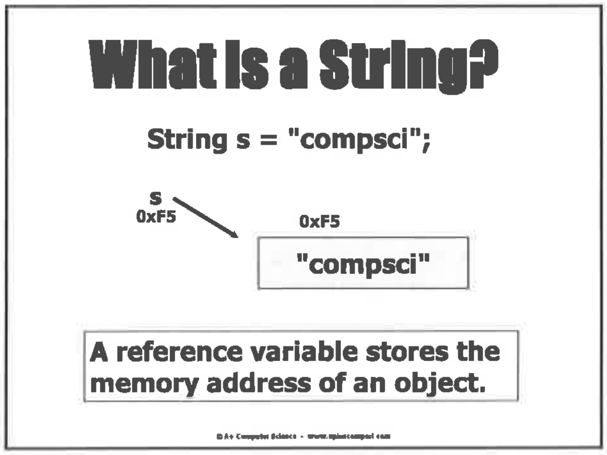


String Constructors

String s = "compsci";
String champ = new String("uilstate");

reference variable
object instantiation

©A+ Computer Science - www.apluscompsci.com

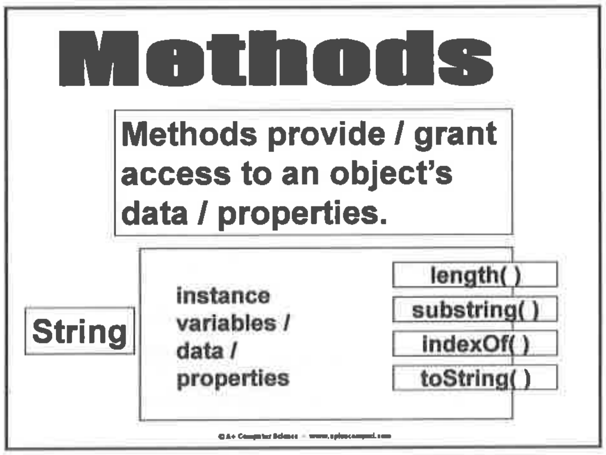


What Is a String?

String s = "compsci";

A reference variable stores the memory address of an object.

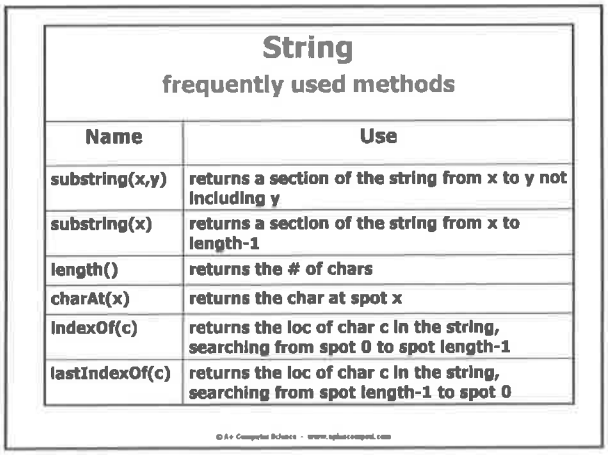
©A+ Computer Science - www.apluscompsci.com



Methods

Methods provide / grant access to an object's data / properties.

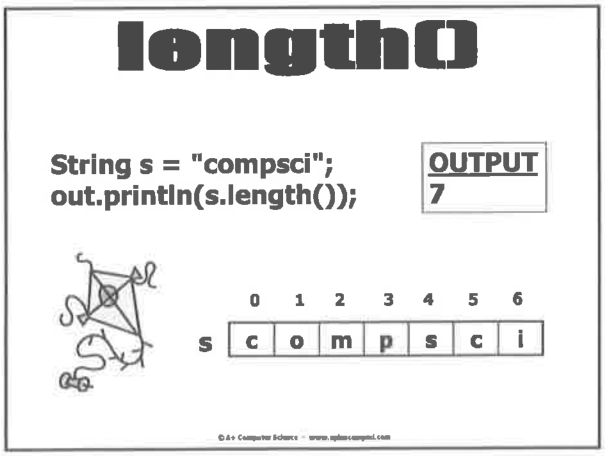
©A+ Computer Science - www.apluscompsci.com



String
frequently used methods

Name	Use
substring(x,y)	returns a section of the string from x to y not including y
substring(x)	returns a section of the string from x to length-1
length()	returns the # of chars
charAt(x)	returns the char at spot x
indexOf(c)	returns the loc of char c in the string, searching from spot 0 to spot length-1
lastIndexOf(c)	returns the loc of char c in the string, searching from spot length-1 to spot 0

©A+ Computer Science - www.apluscompsci.com



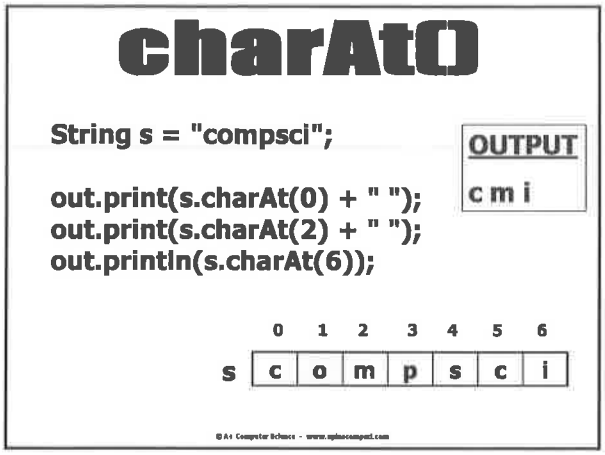
length()

```
String s = "compsci";
out.println(s.length());
```

OUTPUT
7

	0	1	2	3	4	5	6
s	c	o	m	p	s	c	i

©A+ Computer Science - www.apluscompsci.com



charAt()

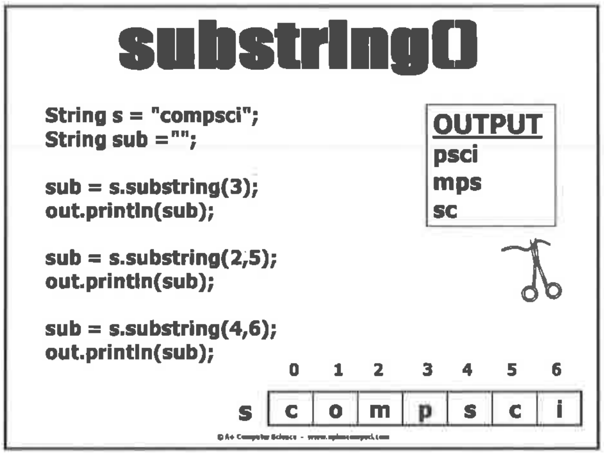
```
String s = "compsci";

out.print(s.charAt(0) + " ");
out.print(s.charAt(2) + " ");
out.println(s.charAt(6));
```

OUTPUT

```
c m i
```

	0	1	2	3	4	5	6
s	c	o	m	p	s	c	i

© A+ Computer Science - www.apluscompsci.com



substring()

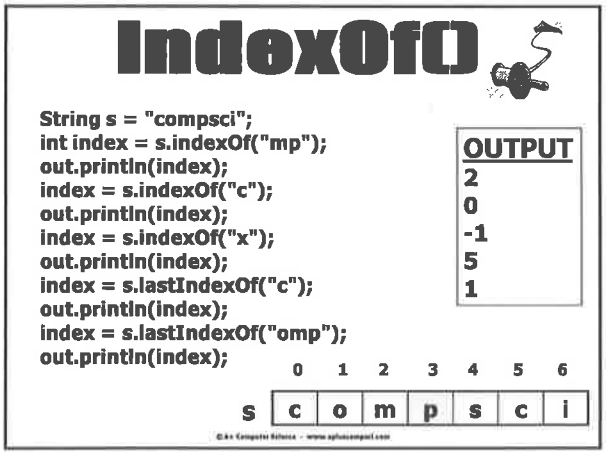
```
String s = "compsci";
String sub ="";

sub = s.substring(3);
out.println(sub);

sub = s.substring(2,5);
out.println(sub);

sub = s.substring(4,6);
out.println(sub);
```

OUTPUT

```
psci
mps
sc
```

	0	1	2	3	4	5	6
s	c	o	m	p	s	c	i

© A+ Computer Science - www.apluscompsci.com



IndexOf()

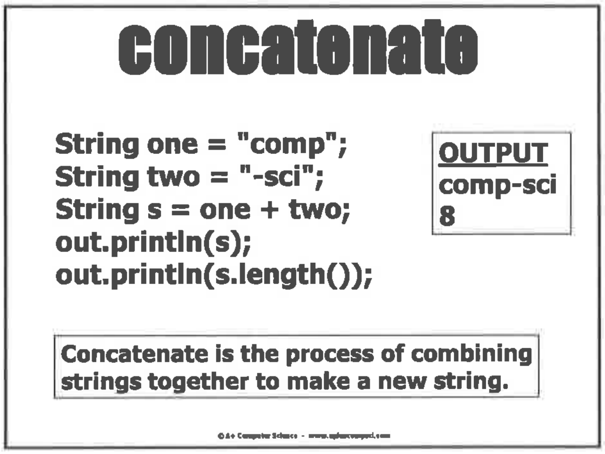
```
String s = "compsci";
int index = s.indexOf("mp");
out.println(index);
index = s.indexOf("c");
out.println(index);
index = s.indexOf("x");
out.println(index);
index = s.lastIndexOf("c");
out.println(index);
index = s.lastIndexOf("omp");
out.println(index);
```

OUTPUT

```
2
0
-1
5
1
```

	0	1	2	3	4	5	6
s	c	o	m	p	s	c	i

© A+ Computer Science - www.apluscompsci.com



concatenate

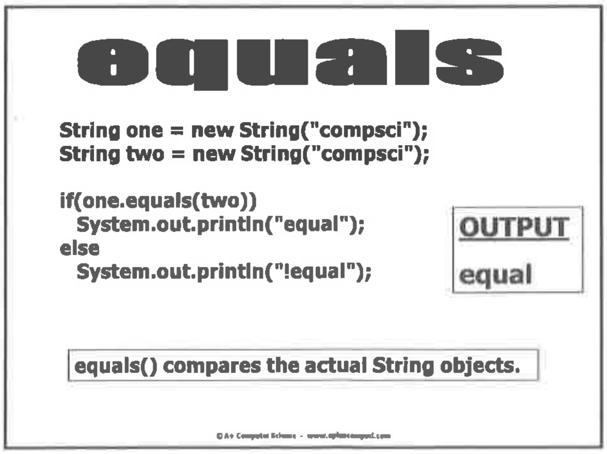
```
String one = "comp";
String two = "-sci";
String s = one + two;
out.println(s);
out.println(s.length());
```

OUTPUT

```
comp-sci
8
```

Concatenate is the process of combining strings together to make a new string.

© A+ Computer Science - www.apluscompsci.com



equals

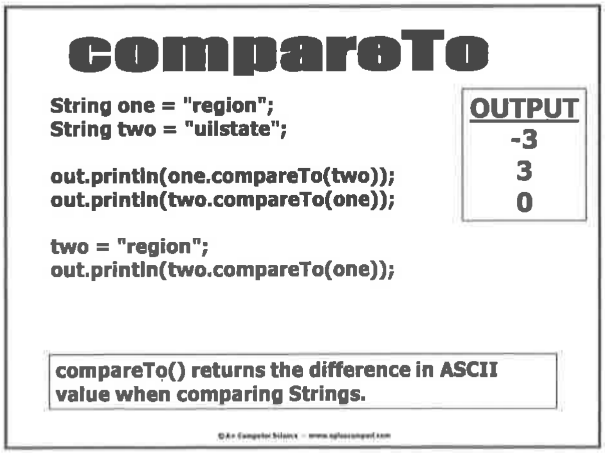
```
String one = new String("compsci");
String two = new String("compsci");

if(one.equals(two))
  System.out.println("equal");
else
  System.out.println("!equal");
```

OUTPUT

```
equal
```

equals() compares the actual String objects.

© A+ Computer Science - www.apluscompsci.com



compareTo

```
String one = "region";
String two = "uilstate";

out.println(one.compareTo(two));
out.println(two.compareTo(one));

two = "region";
out.println(two.compareTo(one));
```

OUTPUT

```
-3
 3
 0
```

compareTo() returns the difference in ASCII value when comparing Strings.

© A+ Computer Science - www.apluscompsci.com