A+ Computer Science

Strings

A Computer Science
www.apluscompsci.com

String

```
String s = "apluscs";
```

	0	1	2	3	4	5	6
s	a	p	l	u	s	c	s

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

String

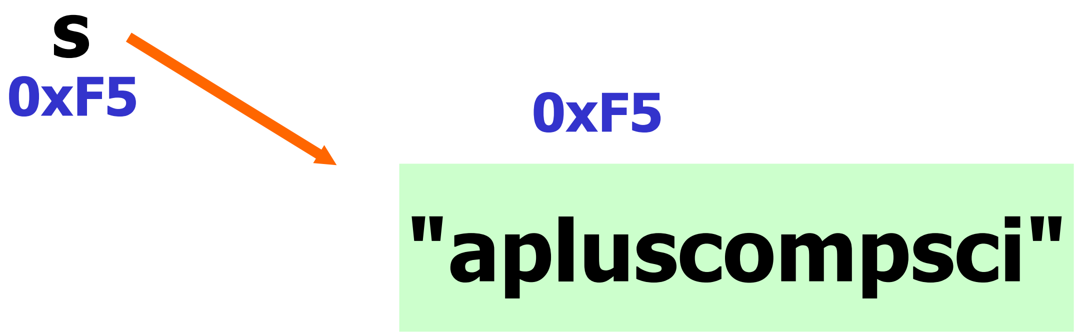
```
String s = "apluscompsci";
String champ = new String("aplus");
```

reference variable

object instantiation

String

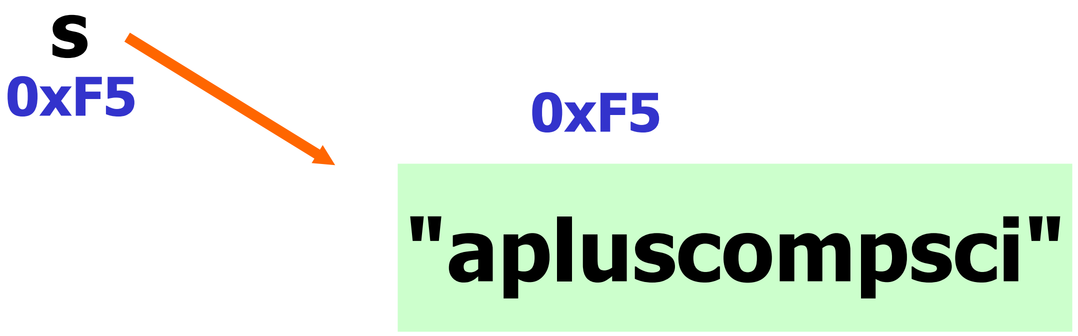
String s = "apluscompsci";



A reference variable stores the memory address of an object.

String

```
String s;
s = new String("apluscompsci");
```



A reference variable stores the memory address of an object.

basics.java

String Objects

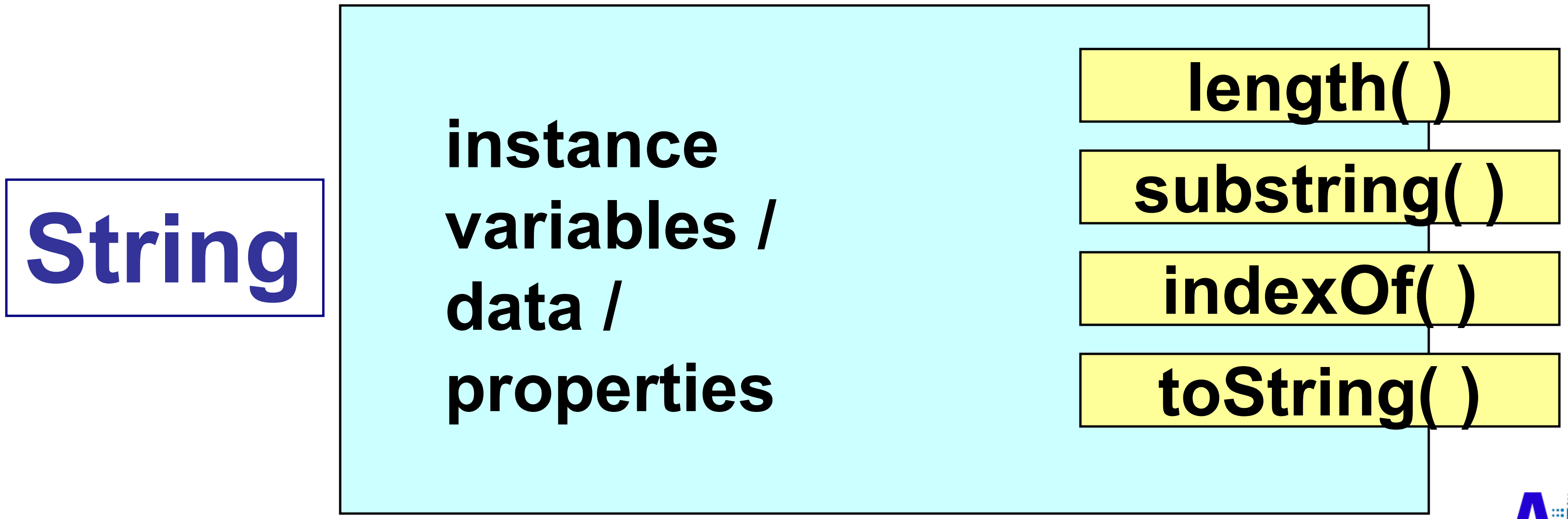
String objects are immutable.

The String class does not contain any modifier / mutator methods.

```
new String("uiltcea");
"statechamps"
"alligator"
```

String Methods

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



String
frequently used methods

Name	Use
charAt(x)	returns the char at spot x
length()	returns the # of chars
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

part of java.lang package

String length()

```
String s = "apluscs";
int len = s.length();
System.out.println( len );
```

OUTPUT

```
7
```

	0	1	2	3	4	5	6
s	a	p	l	u	s	c	s

String length()

Return methods perform some action and return a result back.
.length() is a return method.

```
String s = "apluscs";
int len = s.length();
System.out.println( len );
```

length() returns an integer back to the calling location.
The value returned is then assigned to variable len.

String charAt()

```
String s = "apluscs";

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

OUTPUT

```
a l s
```

	0	1	2	3	4	5	6
s	a	p	l	u	s	c	s

length.java
charat.java

String substring()

```
String s = "apluscs" , sub ="";

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

sub = s.substring(0,3);
out.println(sub);

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

OUTPUT

```
uscs
apl
scs
```

	0	1	2	3	4	5	6
s	a	p	l	u	s	c	s

String substring()

```
String s = "apluscs", 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

```
uscs
lus
sc
```

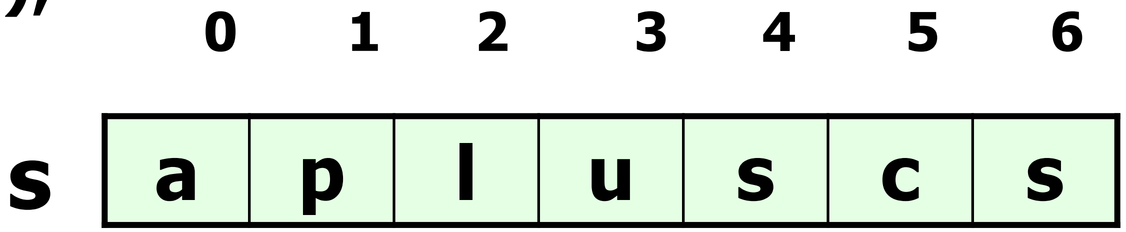
	0	1	2	3	4	5	6
s	a	p	l	u	s	c	s

String substring()

```
String s = "apluscs", sub ="";

sub = s.substring(0,1);
out.println(sub);

sub = s.substring(1,2);
out.println(sub);

sub = s.substring(2,3);
out.println(sub);
```

OUTPUT

```
a
p
l
```



substring.java

String
frequently used methods

Name	Use
indexOf(str)	returns the loc of String str in the string, searching from spot 0 to spot length-1
indexOf(ch)	returns the loc of char ch in the string, searching from spot 0 to spot length-1
lastIndexOf(str)	returns the loc of String str in the string, searching from spot length-1 to spot 0
lastIndexOf(ch)	returns the loc of char ch in the string, searching from spot length-1 to spot 0

part of java.lang package

String indexOf()

```
String s = "apluscs";
int index = s.indexOf("us");
out.println(index);
index = s.indexOf("c");
out.println(index);
index = s.indexOf('x');
out.println(index);
```

OUTPUT

```
3
5
-1
```

	0	1	2	3	4	5	6
s	a	p	l	u	s	c	s

String indexOf()

```
String s = "apluscs";
int index = s.indexOf("pl");
out.println(index);
index = s.lastIndexOf('c');
out.println(index);
index = s.lastIndexOf("plus");
out.println(index);
```

OUTPUT

```
1
5
1
```

	0	1	2	3	4	5	6
s	a	p	l	u	s	c	s

indexof.java

Complete Method Chunk

```
/*
 *method getFirstChunk() should return
 *all letters up to the first @ sign
 *if there is no @ return "aplus"
 *if the string starts with an @, return "APLUS"
 */

public static String getFirstChunk( String line )
{
    return "";
}
```

chunk.java

Work on Programs!

Crank Some Code!

String
frequently used methods

Name	Use
equals(s)	checks if this string has same chars as s
compareTo(s)	compares this string and s for >,<, and ==

part of java.lang package

The equals( ) method

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


System.out.println( one.equals(two) );

System.out.println( one.equals("comp") );
```

OUTPUT

```
true
false
```

equals() compares the values stored in the actual String objects.

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
```

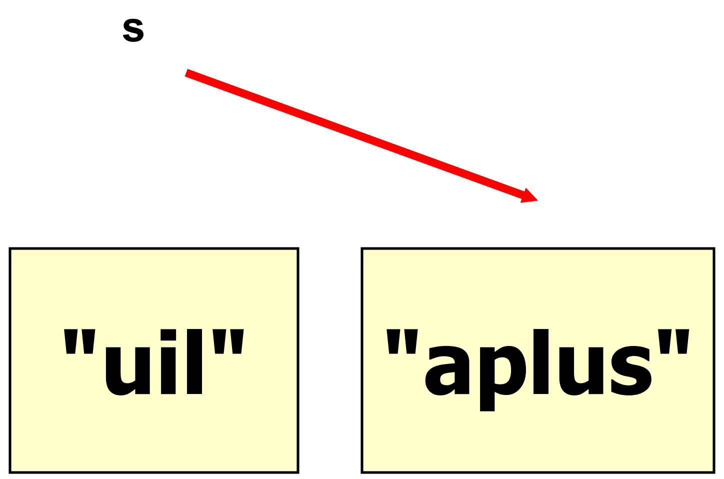
compareTo() returns the difference in ASCII value when comparing Strings.

compare.java

String References

A String reference variable can be changed, but the String object the variable refers to cannot be changed.

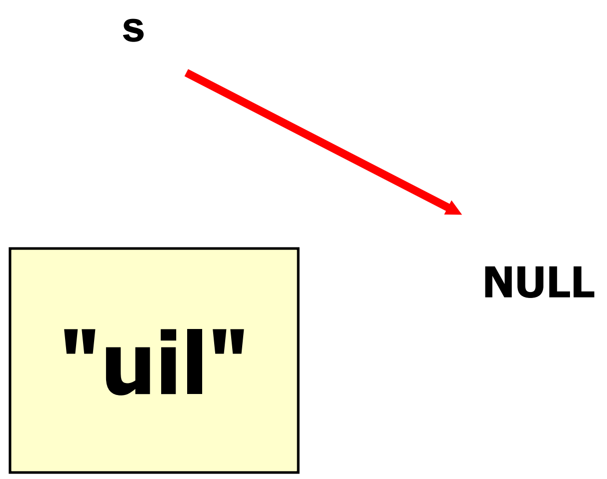
```
String s = "uil";
out.println(s);
s = "aplus";
out.println(s);
```



String References

A String reference variable can be changed, but the String object the variable refers to cannot be changed.

```
String s = "uil";
out.println(s);
s = null;
out.println(s);
```



String References

A String reference variable can be changed, but the String object the variable refers to cannot be changed.

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

out.println(s);
```

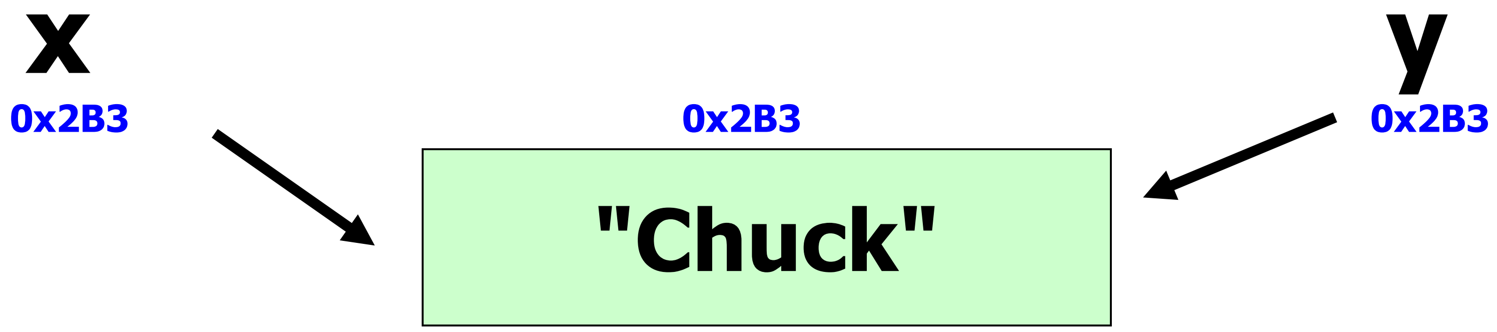
OUTPUT

```
compsci
compsci
COMPSCI
```

References

```
String x = new String("Chuck");
String y = x;
```

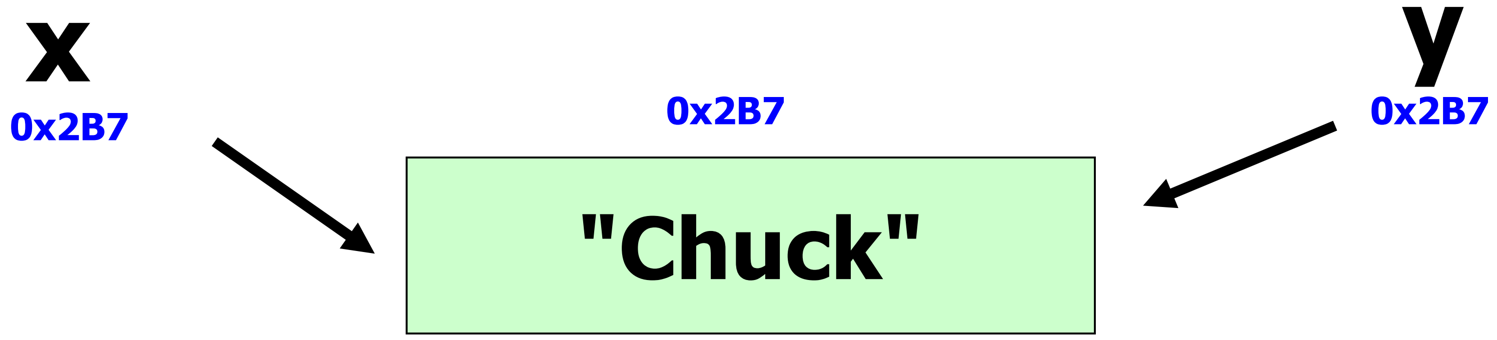
x and y store the same memory address.



References

```
String x = "Chuck";
String y = "Chuck";
```

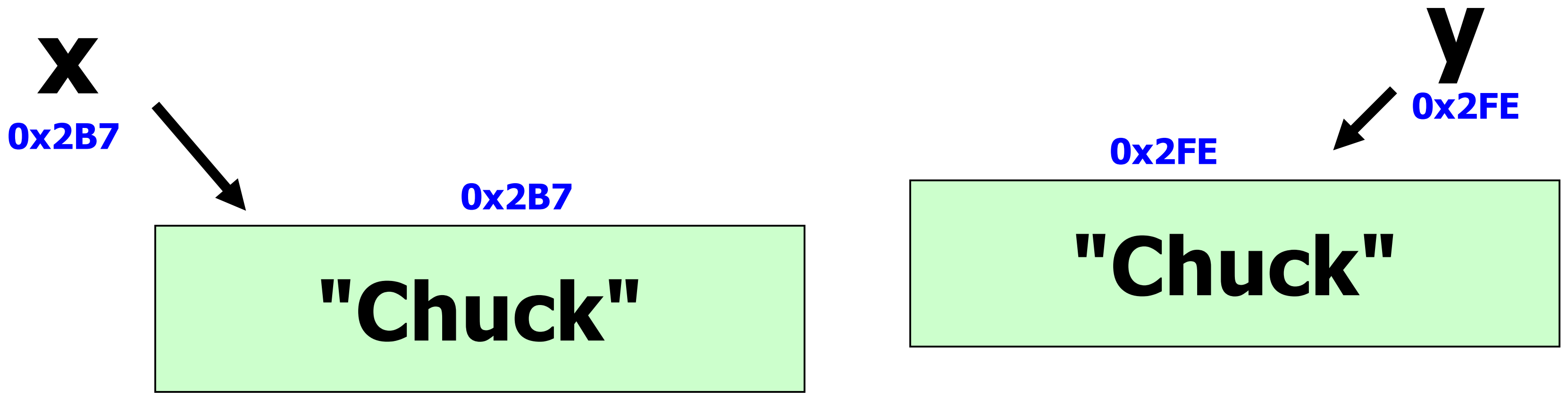
x and y store the same memory address.



References

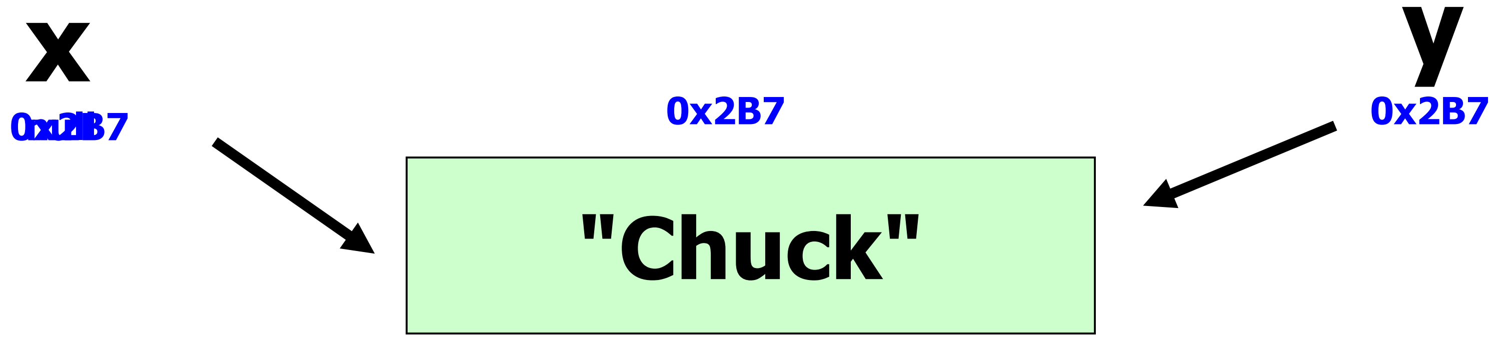
```
String x = new String("Chuck");
String y = new String("Chuck");
```

x and y store different memory addresses.



References

```
String x = "Chuck";
String y = "Chuck";
x = null;
```



touppercase.java

string_references.java

Concatenate

```
String one = "apluscomp";
String two = "-sci";
String s = one.substring(0,4) + two;
out.println( s );
out.println( s.length() );
```

OUTPUT

```
aplu-sci
8
```

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

Concatenate

```
String one = "aplus";
one = one + 7;
System.out.println( one );
out.println(one.length());
```

OUTPUT

```
aplus7
6
```

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

Concatenate

```
String one = "it";
Double x = 99.5;
one = one + x;
System.out.println( one );
out.println(one.length());
```

OUTPUT

```
it99.5
6
```

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

concatenate.java

A Computer Science
www.apluscompsci.com

Parsing Strings

```
int i =
i = Integer.parseInt("2343");
out.println( i );
```

OUTPUT

```
2343
23.78
```

```
double d = Double.parseDouble("23.78");
out.println( d );
```

stringtonums.java

APIs

An API is a collection of prewritten classes and code that
can be used to write programs.

The String class is part of the java.lang package.

Work on Programs!

Crank Some Code!

A+ Computer Science

Strings