

Strings

Return

Methods

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.**

String Constructors

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

```
String champ = new String("uilstate");
```



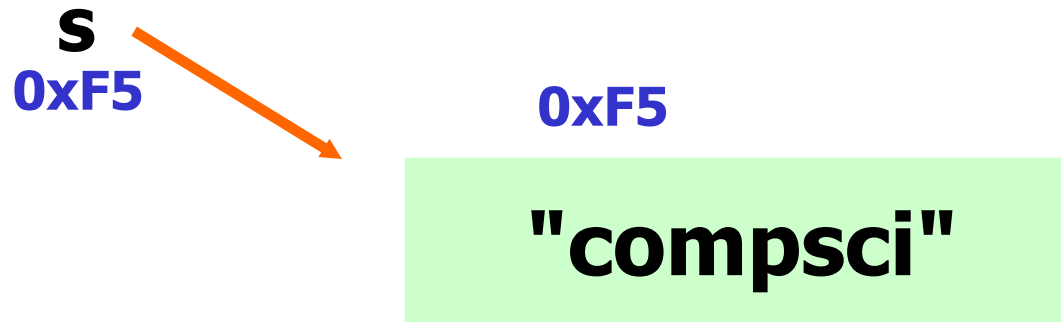
**reference
variable**



**object
instantiation**

What is a String?

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



A reference variable stores the memory address of an object.

Methods

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

String

instance variables / data / properties

length()

substring()

indexOf()

toString()

String

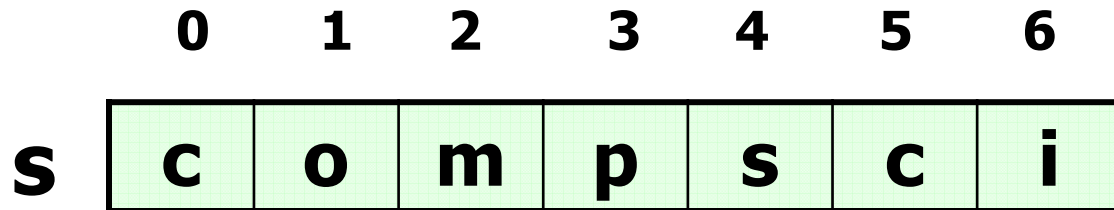
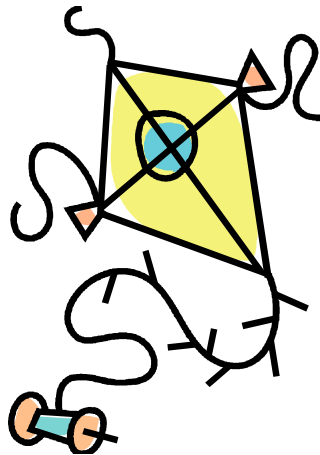
frequently used methods

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

length()

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

OUTPUT
7



Return Methods

Return methods perform some action and return a result back.

`.length()` is a return method.

```
String s = "compsci";  
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.

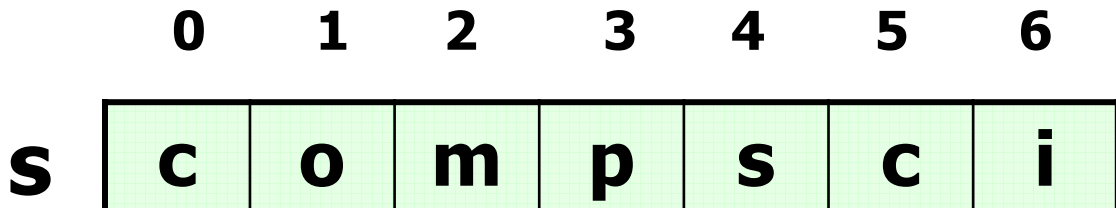
charAt

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

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

OUTPUT

c m i



substring()

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

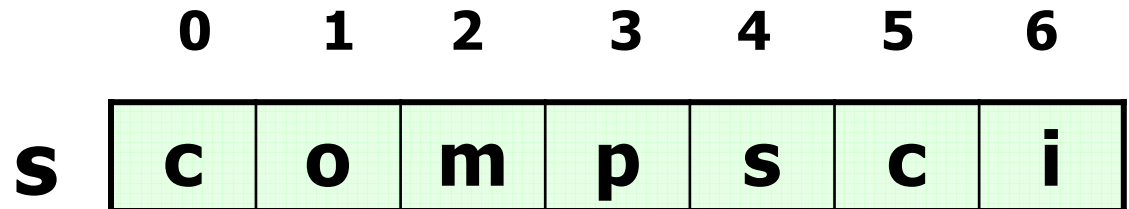
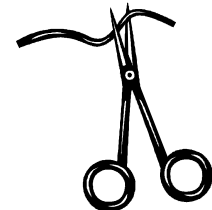
```
sub = s.substring(3);  
out.println(sub);
```

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

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

OUTPUT

psci
com
sci



substring()

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

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

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

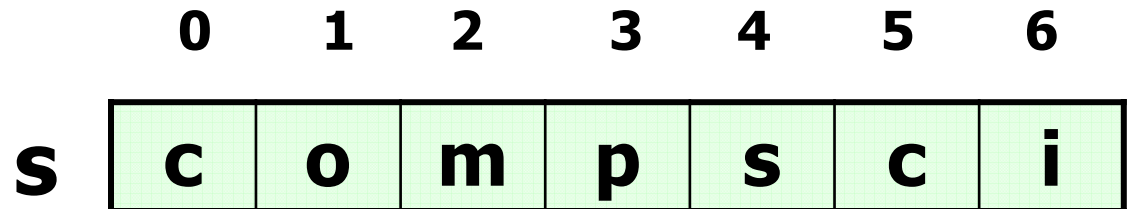
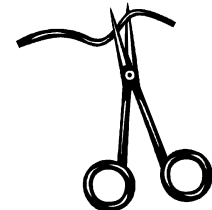
```
sub = s.substring(4,6);  
out.println(sub);
```

OUTPUT

mpsci

mps

sc



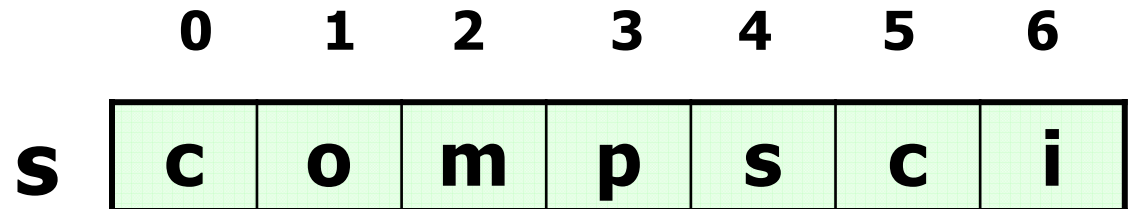
indexOf



```
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);
```

OUTPUT

```
2  
0  
-1
```



indexOf



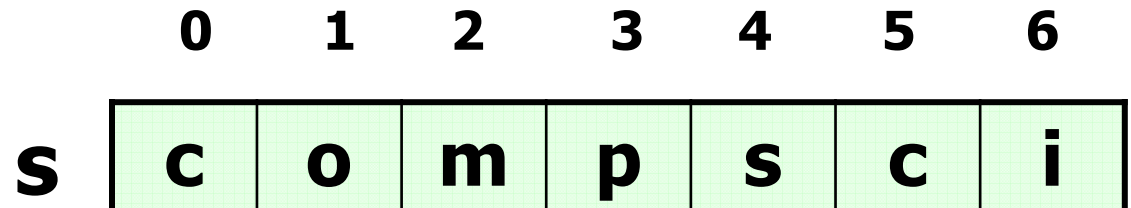
```
String s = "compsci";  
int index = s.indexOf("pm");  
out.println(index);  
index = s.lastIndexOf("c");  
out.println(index);  
index = s.lastIndexOf("omp");  
out.println(index);
```

OUTPUT

-1

5

1



concatenate

```
String one = "computer";  
String two = "-sci";  
String s = one.substring(0,4) + 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.

**return
methods
expanded**

Return Methods

Return methods perform some action and return a result back to the **calling location**.

```
int num = keyboard.nextInt();
```

nextInt() returns an int back to the calling location.

The value returned is assigned to num.


Return Methods

```
Scanner keyboard =  
    new Scanner(System.in);
```

```
int num = keyboard.nextInt();  
out.println(num);
```

num
1

return
method



INPUT
1

OUTPUT
1

Return Methods

```
Scanner keyboard =  
    new Scanner(System.in);
```

```
double num = keyboard.nextDouble();  
out.println(Math.ceil(num));
```

num
3.45

return
methods

INPUT
3.45

OUTPUT
4.0

Return Methods

```
public class ReturnOne
{
    public int twice( int x ) //this is a return method
    {
        return 2*x;
    }
}
```

//code in the main of another class

```
ReturnOne demo = new ReturnOne();
out.println(demo.twice(25) );
out.println(demo.twice(17) );
```

OUTPUT

50

34

Return Method

access

return type

name

params

code

```
public          int          twice( int x )  
{  
    return 2*x;  
}
```

toString

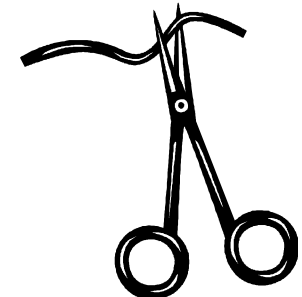
```
class Triangle
{
    private int sideA, sideB, sideC;

    public Triangle(int a, int b, int c)
    {
        sideA=a;
        sideB=b;
        sideC=c;
    }

    public String toString()
    {
        return sideA + " " + sideB + " " + sideC;
    }
}
```

return type

return method



constructors

```
public Triangle()
```

```
{
```

```
    sideA=0;
```

```
    sideB=0;
```

```
    sideC=0;
```

```
}
```

**Default
Constructor**

**Constructors are similar to methods.
Constructors set the properties of an
object to an initial state.**

constructors

```
public Triangle(int a, int b, int c)
```

```
{
```

```
    sideA=a;
```

```
    sideB=b;
```

```
    sideC=c;
```

```
}
```

Initialization

Constructor

**Constructors are similar to methods.
Constructors set the properties of an
object to an initial state.**

modifier methods

```
public void setSides(int a, int b, int c)
{
    sideA=a;
    sideB=b;
    sideC=c;
}
```

Modifier methods are methods that change the properties of an object.

accessor methods

```
public int getSideA()  
{  
    return sideA;  
}
```

Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.

accessor methods

```
public String toString()  
{  
    return "" + getSideA() + " " + sideB + " " + sideC;  
}
```

Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.

encapsulation

All data members should have private access. The public constructors, accessor methods, and modifier methods should be used to manipulate the data. All data is tucked away nicely inside the class.

encapsulation

The public methods give you access to an object's private data / properties.

**Class/
Object**

private data /
instance variables /
properties

getIt()

setIt()

toString()