

Something **new**: String  
class odds & ends

Mr. Neat  
Java

# String Details: Constructing Strings

you don't have to use *new*

// formal way to construct a String object

```
String a = new String("happy");
```

// but this literal method works also

```
String b = "ice cream";
```

# String Details: Strings are Immutable!

Every time you use String var, a new String object is made:

```
// makes and stores a String object in RAM
```

```
String c = new String("happy");
```

```
// makes and stores another String object in RAM
```

```
String d = "ice cream";
```

```
// makes and stores another String object in RAM
```

```
d = "bananas";
```

# String Details: Convert an int to a String...

```
int e = 27;  
// f is the String "27"  
String f = ""+e;  
// note the above uses the String concatenate operator  
// as a review  
String g = "I love ";  
String h = "pizza"  
System.out.println(g+h);  
// the above line outputs I love pizza
```

# String Details: Escape Sequences

// \n is called an escape sequence which inserts a new  
// line into the string

// for example:

```
String j = “I like \n happy face pancakes!”;  
System.out.println(j);
```

output:

I like

happy face pancakes!

# String Details: Escape Sequences

// \t is called an escape sequence which inserts a tab into the string

// for example:

```
String j = "I like \t happy face pancakes!";  
System.out.println(j);
```

output:

```
I like    happy face pancakes!
```

# String Details: Escape Sequences

// \" is called an escape sequence which inserts a “ character into the string

// for example:

```
String j = “I like \" happy face pancakes!”;  
System.out.println(j);
```

output:

I like “ happy face pancakes!

# String Details: Escape Sequences

// \\ is called an escape sequence which inserts a \ character into the string

// for example:

```
String j = "I like \\ happy face pancakes!";  
System.out.println(j);
```

output:

I like \ happy face pancakes!



# Lab