Primitive Types Recap and the String Data Type

In the previous video, we looked at the **char**, and also the **boolean** types, which were Java's seventh and eighth data types.

So at this point, you should be familiar with all eight of Java's primitives.

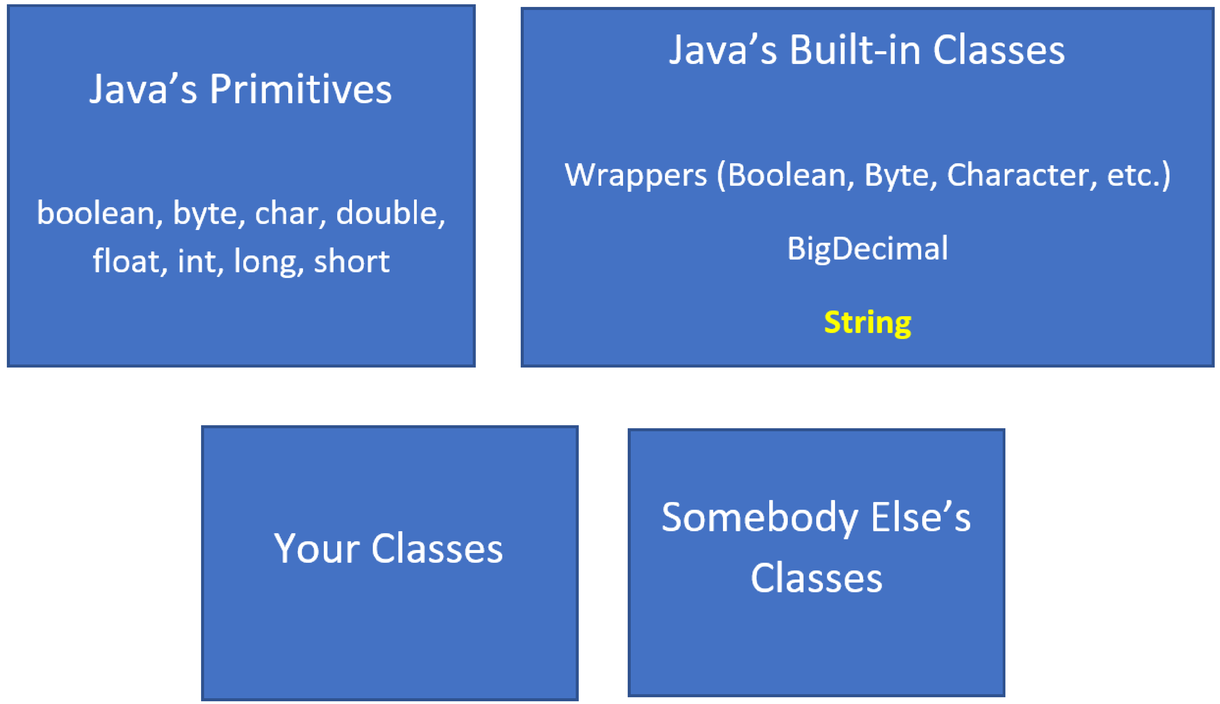
Java's 8 Primitive Data Types

Whole number	**Real Number (floating point or decimal)**
byte short **int** long	float **double**
Single character	**Boolean value**
char	boolean

The int and a double are Java's default data types for numeric literals

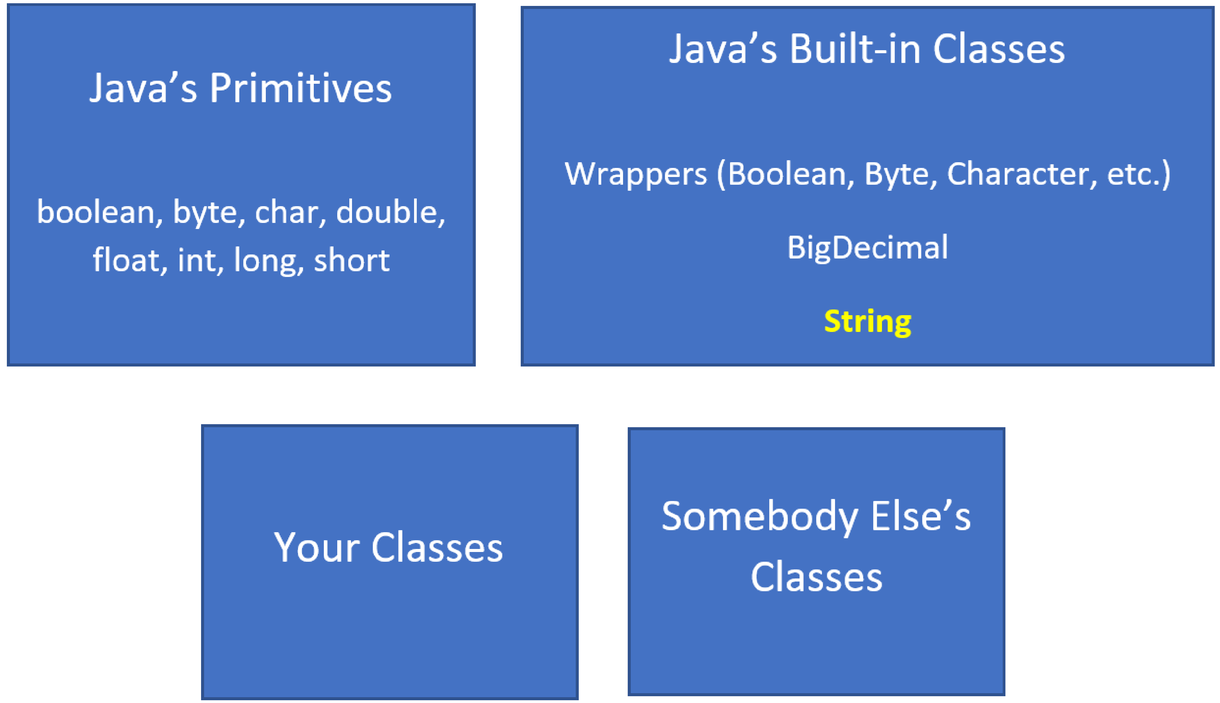
Handling Data in Java

This slide demonstrates that most Java programs use some combination of the data types shown in this diagram.



Remember, I've said that classes in Java are custom data types

Handling Data in Java



Remember, I've said that classes in Java are custom data types

You'll use Java's primitive data types, Java's built-in classes, and probably some combination of your own custom classes and somebody else's.

So What is a String?

A String is a class that contains a sequence of characters.

Executing multiple lines of code in JShell

To execute multiple lines of code as a set, in JShell, first start with an opening curly brace and press enter.

```
jshell> {        // start with a curly opening brace
   ...>     first_statement;
   ...>     second_statement;
   ...>     third_statement;
   ...> }        // end with a curly closing brace
```

JShell will display an alternate prompt as you can see, three dots and a greater than sign.

You can add a statement and press enter, until you've added as many statements as you want to run.

Finally, add the closing curly brace, noting that a semicolon is not required after this brace.

Once you press enter after the closing brace, all of your statements will run in the order you put them.

Executing Multiple Statements In JShell

There are two ways to execute multiple statements in JShell:

- Put your statements on a single line.
- Or, enclose your statements in a set of curly braces {}.

String concatenation

In Java, the + symbol is an operator which can mean addition, if used for numbers.

But it also means concatenation when applied to a String.

A String + anything else, gives us a String as a result, concatenating anything after the String as text to the initial String.

Strings are Immutable

Immutable means that you can't change a String after it's created.

So in the case of the code we've written, the value 120.47 is technically not appended to the current contents of **lastString**.

```
lastString = lastString + myInt;
```

Instead, a new String is created automatically by Java. The new String consists of the previous value of **lastString**, plus a textual representation of the double value 120.47.

The net result, is that our variable, **lastString**, has the concatenated value. However Java created a new String in the process, and the old one will get discarded from memory automatically.

String vs StringBuilder

- The String class is immutable, but can be used much like a primitive data type.
- The StringBuilder class is mutable, but does not share the String's special features, such as being able to assign it a String literal or use the + operator on it.

Both are classes, but the String class is in a special category in the Java language.

The String

- The String is so intrinsic to the Java language, it can be used like a 9th primitive type.
- But it's not a primitive type at all, it's a class.