String Literal Example

If you recall, we've used literal strings before, and that's where we've typed some text in double quotes.

```
jshell> System.out.print("Hello World");
```

We haven't really used a String variable yet, but we'll be doing that in upcoming videos.

Comparing the char to the String

This table is a quick summary of the differences between the char and the String.

char	String
• Holds one, and only one, character • Literal enclosed in Single Quotes	• Can hold multiple characters • Literal enclosed in Double Quotes

Is there a good use for the char data type in today's computing world?

Why would you want to use a variable that only allows you to store one character?

One example might be to store the last key pressed by a user in a game.

Another example might be to loop programmatically through the letters in an alphabet.

char Data Type

A char occupies two bytes of memory, or 16 bits, and thus has a width of 16.

The reason it's not just a single byte, is that a char is stored as a 2 byte number, similar to the short.

This number gets mapped to a single character by Java.

- So, when you print a char, you will see the mapped character, and not the representative number.
- And you can use single quotes and a character literal to assign a value to a char, which is much simpler than looking up the representative number.

Unicode

Unicode is an international encoding standard for use with different languages and scripts by which each letter, digit, or symbol is assigned a unique numeric value that applies across different platforms and programs.

In the English alphabet, we've got the letters A through Z, meaning only 26 characters are needed in total to represent the entire English alphabet.

But other languages need more characters, and often a lot more.

Assigning values to a char variable

There are three ways to assign a value to a char: Each of these methods, represents storing the letter, capital D, in memory.

Assignment Type	Example Code
a literal character	`char myChar = 'D';`
a Unicode value	`char myChar = '\u0044';`
an integer value	`char myChar = 68;`

The char Challenge

Create three char variables to store the character for the question-mark symbol

- `mySimpleChar` should be assigned the literal question-mark character ?.
- `myUnicodeChar` should be assigned the unicode value for the question-mark ?.
- `myDecimalChar` should be assigned the decimal value for the question-mark ?.
- Print all three variables in one statement, that starts with the label 'My values are '.

Hint: Use the chart at www.unicode-table.com

Note on JShell and UTF-8 unicode values

There are a lot more characters, which are not on the usual keyboard, that can be output by this method, for example the copyright symbol.

But, if you are testing this out on your own, using Windows, you should be aware that JShell may give you an unexpected result, because UTF-8 is not supported, by default, for command line operations.

I point it out here, in case you are being adventurous, and do encounter this problem.

Again, this is only a problem with JShell and Windows users.

This won't be a problem in Java or IntelliJ, or if you're using MAC or Linux.

Boolean Primitive Type

A boolean value allows for two opposite choices, true or false, yes or no, one or zero.

In Java terms, we've got a boolean primitive type, and it can be set to two values only, either true or false.

The wrapper for boolean is Boolean with a capital B.

Why would you start your boolean variable name with the prefix 'is'?

Developers will often use the word, is, as a prefix for a boolean variable name.

This creates a name that seems to ask a question, which makes reading the code more intuitive.

But other prefixes can be just as valid.

Why would you start your boolean variable name with the prefix 'is'?

Here are some example boolean variable names, such as isMarried and hasChildren, that clearly define what condition is being tested:

Boolean variable name examples
isCustomerOverTwentyOne
isEligibleForDiscount
hasValidLicense
isMarried
hasChildren