Ternary Operator

In previous videos in this section, we've learned how to use the 'if then' statement, as well as experimenting with the Logical **and**, and the Logical **or** operators.

In this video, we're going to take a look at something called the Ternary Operator.

The Ternary Operator (Condition ?: Operator)

The ternary operator – Java officially calls it the Conditional Operator – has three operands, the only operator currently in Java that does have three.

The structure of this operator is:

```
operand1 ? operand2 : operand3
```

Ternary Operator ? :

The **ternary** operator is a shortcut to assigning one of two values to a variable, depending on a given condition.

So think of it as a shortcut of the **if-then-else** statement. So far in the course, we've only discussed if-then and not else. I'll be discussing else in the next section when we go deeper into control blocks.

Ternary Operator ? :

Consider this example:

```
int ageOfClient = 20;
String ageText = ageOfClient >= 18 ? "Over Eighteen" : "Still a kid";
System.out.println("Our client is " + ageText);
```

Operand one – **`ageOfClient >= 18`** in this case is the condition we're checking. It needs to return `true`, or `false`.

Operand two – **"Over Eighteen"** here is the value to assign to the variable **ageText** , if the condition above is `true`.

Operand three – **"Still a kid"** here is the value to assign to the variable **ageText** , if the condition above is `false`.

Ternary Operator ? :

```
int ageOfClient = 20;
String ageText = ageOfClient >= 18 ? "Over Eighteen" : "Still a kid";
System.out.println("Our client is " + ageText);
```

In this particular case, ageText is assigned the value "`Over Eighteen`", because `ageOfClient` has the value 20, which is greater than or equal to 18.

Now it can be a good idea to use parentheses, like this example below, to make the code more readable, particularly in the ternary operator.

```
String ageText = (ageOfClient >= 18) ? "Over Eighteen" : "Still a kid";
```

Ternary Operator ? :

In the first example we looked at in our code, we returned a boolean value from the ternary operation.

```
boolean isDomestic = makeOfCar == "Volkswagen" ? false : true;
```

This was a good way to demonstrate the ternary operator, but wouldn't be something you'd do when writing proper code.

A much simpler way to write this code is shown here:

```
boolean isDomestic = (makeOfCar != "Volkswagen");
```

You can see that this code has the same effect and is quite a bit easier to read.