Transcript: Ternary Operator

>> Now let's take a look at the conditional or ternary operator. It has the following form and you can think of it like a concise if-else, but it's an expression rather than a statement. So it begins with a boolean expression, followed by a question mark. And then, an expression that you do if it's true followed by a colon.

And then an expression that you do if it's false. So, there's three operands if you will separated by a question mark and colon. And the three operands sort of suggest the name ternary operator. That's why it's, ternary is three, of course. So, let's look at some examples. Here's one here, we wanna print Right or Wrong, depending on isCorrect.

IsCorrect is a boolean variable, and so we've got a System.out.print line, isCorrect. If it evaluates to true, it returns the, this whole expression, it evaluates to Right. If isCorrect is false, the expression evaluates to Wrong. Here's another one. Here we want to subtract a, a discount, which is a double, from price, also a double if discount is above 0.

And so, here we've got double total, and we're assigning that to this expression. And the expression, of course, is our ternary operator. So we've got discount greater than 0, if that evaluates to true, then this expression is gonna return price minus discount. If it evaluates to false, it's just gonna turn, return price.

And so as long as discount is greater than zero, we're gonna give the, the customer a discount. If the discount's less than zero, we don't, of course. And here's a third example here we wanna print unit or units based on the value of unit. And here, we're concerned about whether units, the number of units is, is more than one or not.

And so here we say System.out.print Total, and this is the variable for units. And then we say this is a string of course units gets converted to a string and gets appended to Total. And, and then our, our ternary operator is here. Here we've got units == 1.

If that's true, then this expression evaluates to unit. Otherwise, it evaluates to units, and so we're gonna either concatenate unit or units depending on whether this is 1 or not. So that's, that's a very useful thing when you're, when you're printing out messages or reports and things like that whether you wanna determine if the variable is plural or not, depending on what it goes with it.

By the way, English is, is strange in that zero is plural. If units was zero, we would say Total is zero units, as opposed to unit. Just sort of an odd thing with English there. Now, when to use the ternary operator it can make a simple if-else statement more concise.

Here we've got isCorrect, print Right else print Wrong. And we just saw that this can be converted to a this concise statement. So this, this is a good time to use it, very, very simple thing to do, and you saved a lot of space up here, and easy to read.

Now, you've got to weigh, though, conciseness, which we had previously right there in that example, versus readability. This ternary operator may make the logic in your code hard to follow. Consider this example here. Here we've got a method that returns the number of small bars needed to reach a goal based on small and large chocolate bars available.

And so here we've got this in a single expression, and so our goal is, is this many bars, and we have this many small and this many big. And we assume that the big yard, big bars will be consumed first, and then we'll put the small ones to go with that.

So this is, this is actually pretty complicated looking, and there, obviously this can be rewritten as an, as an, as an if-else. But let's take a look at the code. So here's our code, and just to help you understand what this problem does a little more, here we're going to make chocolate with 4, 1, and 9, so we're sending in 4 small bars, 1 large, and our goal is 9.

Well clearly one large bar, which is five pounds in this case, looking up here. In fact, here we've got the documentation, the small bar, think of it as one pound, and a big bar is 5 pounds. So these are big bars of chocolate. So here if we send it we want nine pounds of chocolate.

Well, that's gonna be 1 large bar and 4 small, we can see that. In fact, let's we'll run this in debug mode, and we'll see it work. So we're, we'll, we'll step in, and it, it calls our method, and returns. And here we print for our 4 small, 1 big, and a goal of 9 we would end up using 4 small.

That is, one big is gonna be five pounds and then we need four small to get to the goal of 9 there. And let's just step through these others. Here's the next one, this is 4, 1, and 10. And notice here, we're gonna return a -1 because we actually can't do this.

We don't have enough chocolate here with one big bar, which is five pounds, and then four one-pounders. And then, finally, we'll send it down here 4, 7, and 1. And here we see that we would use 2 of the small bars. So one big bar would be 5 and 2 small bars would make 7, so we would reach our goal of 7 there.

So program works, but the, the real issue is whether you would do this or not. Is is the complexity of the ternary operator expression. So this has actually nested ternary operators in it. And likely it would be easier to read if this was an if-else type situation. So I'll leave that to you as an exercise but now you know how to use the ternary operator.