Transcript: *for* Statement– ArrayList

>> Let's see how we could use the for statement with an ArrayList. Suppose that list is an ArrayList holding names of type String, and that we wanted to print out each name. We could use the following code, we could say for (int i = 0; as long as i < list.size.

We're gonna go through the loop and increment i on each iteration. Notice in the loop here we're just saying list.get(i) and, and we're printing that out. Now it turns out that the ArrayList is Iterable and, and there's a special statement called a for-each, or enhanced for loop that we can use to loop through an ArrayList.

And here's the statement for that, we've got for, String name, and list. We wannna print out name. Now this is a special loop, and, notice here we're, we're just gonna go through the list and we don't have any, indexes set or, or, or Boolean. We're just gonna start with the, first thing in the list and loop all the way through it.

And we always read the header. When you see this format, you wanna read the header as foreach String name in the list. So just to reiterate here, the first item there, in the for header, in this for-each, loop or enhanced var is, the type of the object in the ArrayList.

And then next, we have the variable that's used to reference the current item, in each iteration. So the first time we go through the list, this'll be the first name in the list if you will, or first String in the list. Second time, it'll be the second one in the list and so on all the way through the list.

So we start at the first one and go all the way through the list there. And then the last variable here of course in the header is the ArrayList itself. So let's look at an example called GroupRoster And here we've got a a final a a constant static final int GROUP_MEMBERS 5.

What we're gonna do is read in five names, and store them in an ArrayList. And so here we've got a, a standard for loop. And so we'll go from int to i = 1 as long as i < GROUP_MEMBERS. And that's 5 here, we could set that to anything, but it's set to 5 right now.

So we'll go through the list and we'll print a name, number and that'll be one more than our index. Because people like to start with one instead of zero, of course. And then we'll get the next line in and we'll add that to the list. So we'll go through that five times there and then we're going to sort the list.

I've, imported java.util.Collections and it has a sort routine that works on ArrayList. So right here in one statement we can say Collections.sort, send it our array list and that'll put the items in order. And then finally here, we use a foreach statement to go through the list and print out each name.

And at the end here we'll just get one and also find the length of it just for good measure. So let's run this in the canvas. I'm gonna move this out of the way just a bit there. So, I'm gonna hit Play. We're down here and it's waiting for the first name.

And so I'm gonna key in, Pat. It goes through the loop, see it looping there, iterating, and let's key in Sam, And how about, Jill. See in the canvas, these are getting added to the list there. And of course, we gotta key in Jack here. And then finally, maybe Joe, OK?

Now this is gonna be the last one I key in. I'm gonna pause the canvas here, so I can single step. So I'm gonna go ahead and hit Enter down here. So this will be the last one that gets added, we're about to add Joe, you'll see when I, when I step.

So Joe gets added and now we're going to be greater than the number of members here. We're only gonna get five people on our list so we'll skip out of the loop. And now we're back to solve the loop. We're gonna call collections.sort and we'll absolutely see these reordered in alphabetical order.

And we see that over here now, it's Jack, Jill, Jo, Pat, and Sam. And now we're going to march through our for-each loop. So the first time through the loop, notice we're gonna do this for String name and list, for each string name and list. And, name hasn't been assigned anything yet, but when we step, name gets assigned Jack.

So Jack is the first thing in the list, and we just print name. And next time through the list, soon as we step in for this next iteration, the name is Jill, which is the second item in the list. And as you see we're just marching right on through here, each time name gets a new value.

So inside the body, we can use name however we want, in this case we're just printing. So there's Pat and then finally, one more time we're gonna go through the list and get Sam. And then the next iteration or next step here, we're gonna leave the loop. And we'll go down here and we'll print the Here we're printing the length of Jo is, Jo happens to be the second item on the list and we're just, we did a get to now and now we're actually getting, whoops, let me move over here a little.

We're gonna do a list.get(2).length and print out the length of Jo, which we see is 2. So that, that ends the the program there, so just to reiterate, we can use the stan, a standard for loop, with ArrayList. And, and there we will give a starting index, a a Boolean expression to exit the loop eventually.

And then our loop counter, and that works great. But if we, if we wanna go through a list once we have a list, and we wanna go through it from beginning to end, we can use a foreach. And this is read for-each, String name and list, and we just go through the list here doing whatever it is we need to do for a list.

So, the foreach has the advantage that of course, we don't have any of these indexes being set, and setting those, of course we could make a mistake. So if you wanna go through a loop from beginning to end, the foreach loop is certainly definitely something you should consider.