Welcome to module 6, **scripting** sublist. So far, we've created scripts where we've extracted information off of our record, but the information that we have been extracting so far are information on the body fields. What if we want to get information off of a related record? So that's what we're going to look at in this module.

So here we're going to cover how you can script standard sublist, either from the client-side or on the server-side. Now the behavior is pretty much the same, except that on the client-side, you need to have your user actually type values in or interact directly with the sublist. But on the server-side, since the user cannot really touch the server or physically go to the server and type values in, you have to virtually do that from your script. So that's what we're going to talk about.

We're also going to deal with how you can add a sublist to any record from a custom record. So for example, if you want to make an association between two records, you can do that by creating a sublist. And also how we can script custom sublists. So just in case you did decide to create a custom record, then attach that to another record that as a sublist, how would you be able to script that.

Let's talk about what a sublist is. So what is a sublist? A sublist is basically a related record that is listed on another record because that record is dependent on that child record. For example, a sales order record would not be able to exist without the items, because if you have a sales order, of course, you need to have items that you will be selling. Or if you have an expense report record, you need to have expense lines that you are expensing for.

Now the items is not actually part of your sales or the record, but is actually a different record. Now in NetSuite, you don't really need to perform a join to make this work, because there is already an inherent association between the two records. Again, a sales order could not really exist without an item record.

So if you open a sales order record, you would see that there is a table inside your record, and that table would be your sublist. So for example, here in your sales order, you have two items, so this would refer to two different items-- or two different item records. So the information on that item record-- for example, the name of the record, the price of the record, or if you have a code for that record, or where that record is located in terms of warehouse, then that information can be displayed on your main record.

Now on a standard record, there are some associations that are already built in. But if you want to associate like another record or one record to another record, that is something that you can also do.

Now if you want to script for a sublist, you need to understand what the different elements of a sublist are. So the elements of a subject-- of course, the name of the sublist. So for example, in this case, this is a purchase order. And in a purchase order record, you can see that there are two sublists.

One would be the expenses sublist, and then the other one is the item sublist. So right now, the item sublist is highlighted, and this is called, of course, the sublist name. And that is basically what you would need if you want to refer to a particular sublist, because as you can see, one record you can have multiple sublists.

Now aside from that sublist, you would also have your sublist field. Now with a sublist, fields are represented as columns. So here you have items. The item name-- that is a sublist field. You have the vendor name, received, billed, quantity, units, and descriptions. So all columns are referred to as a sublist field.

And it is represented as a column, not an individual field because you can have multiple items or multiple lines inside your item. So for example, here, we can see that there are two items related to this particular purchase order. You can identify a particular item on your sublist based on a sublist line.

For example, if you want to refer to the Mondrian rug, then you have to make sure you refer on your script to line 0. And also the line numbers always start with a zero similar to an array. So if you want to refer to the first item on your sublist, you have to make sure you set that to line 0, then line 1, line 2, line 3, and so on.

So there you go. So those are the things that you have here. So if you want to get the value of a particular item-- so you have to refer to the sublist name. So you have to put that in your script. So I'm trying to script for, say, the item sublist, I want to get, say, the quantity field, and then I want to get that on the first line, or line number one.

By the way, lines are also referred to as a line item. So when you say line item, it doesn't necessarily mean that it's for the item record. Line item simply means just one line. So it can refer to an item, it can refer to an expense, it can refer to anything. So as long as it's one collection of fields or one record, that is called a line item.

Moving on to the next section of our discussion, we're going to talk about how we can start automating sublists. Let's go ahead and look at how you can script a sublist. So once we know the idea of sublists-- one thing you have to remember, by the way, is that like any record in NetSuite, we don't really need to perform a join.

Now if you need to get, say, item information regarding a particular item on a sales order, just go ahead and grab the sales order, or all of the related item information would already be there, so you don't need to worry about doing a join. And if you're dealing with, say, a SQL table, you have to make sure that you join based on the right ID so you have your keys. And if the keys are not matching, of course, that would mess up your table, and that's something that you have to think about.

But in SuiteScript, just grab the record. If there's anything related to that record, it'll automatically be pulled in. Now that said, going-- or discussing the different methods that you can use for a sublist, you can get that from the record object. So if you have already grabbed the record object either on the client-side script if you're using context dot new record, or context dot current record, or on a server-side script, you're seeing record dot new record, you can just use these functions once you have that object.

So here, the first one we have an example of getLineCount. So getLineCount would be if you want to count the number of line items on a sublist. So if you want to know how many items are there in this particular order, or how many expenses are there in this expense report, you can use the getLineCount.

You also have to get sublist value. So the getSublistValue is similar to getValue. The difference with getValue is, of course, the getSublistValue would specifically look at the sublist to get the value. So that means that you also have more parameters that you need to pass.

So for example, here in this statement, you see this is a setSublistValue-- so instead of getting the value or setting the value here. So the additional lines that you have here is that you have your sublist ID, which is the name of your sublist, and also the line number.

So if you're trying to get the setSublistValue, you need to make sure you have all three. And if you're using getValue for record, if you're dealing with a body field, you only need to put in one which is the field ID. Now in this case, since you're setting a value, you also need to have an additional property here where you can pass in any value that you want to set the currency to.

So same thing here. So aside from get value, set value, you also have set text and get text. But then other than that, you also have special functions or special methods that you can use for setting the sublist values. For example, for the first few methods here, we can see that this would require a line number on the parameter.

So for example, for setSublistValue, you need to include which particular line do you want to set or which particular line do you want to get. But in these last four methods, you can actually dynamically use this so that it would move depending on what the user is currently manipulating.

For example, if the user is manipulating the first line, then line number would automatically be line number 0. If the user is currently on line 5, the line number would be automatically 5. So this would be very useful if you are dealing with a script where you don't really know where the user is going, like maybe the user is on the first line, or the third line, or 10th line on your sublist. So these are the different functions that you can use.

Now one of the problems that you might encounter if you are dealing with a sublist is that, where do I get the ID? Now unlike a body field where you can go to the form, and then you can click on the field-level help, then you would be able to get the ID what would a sublist what's going to happen is that if you try to click on, say, the sublist field which is the column header, it would probably sort that column or sort the list of sublist based on that field. So it's not going to give you a field-level help.

So to get the IDs, what you can do is that you can go to the SuiteScript records browser. Now the SuiteScript records browser is a page or a tool that you can use to get any IDs, not just sublist. You can use that for body fields, you can use that for sublist, you can use that for searching, and so on.

Now if you want to get the URL for the SuiteScript records browser, you can go to working with the SuiteScript browser. So what you can do is that you can pin that to your browser or save that as your favorite. Now regarding the SuiteScript records browser, that would be fairly static, and would only change if NetSuite has upgraded to the next version. That's the only time that it would change.

The older version of the records browser would still be available, though. So for example, you still want to use a certain version of NetSuite or a certain version of SuiteScript, like if you want to-- instead of moving to the latest version, then you can use this as a reference.

If I want to move to another version-- let's say I want to move to 2015.2 version, all I need to do is just change the number. So it should say URL, and then when it gets to the browser section, you can just say browser 2015 underscore 2. So you can do that, and if you change the version number, it should move to the previous version.

That normally isn't something that you would do, because you would almost always just use the latest version, but that is something that you can if you want to. But developers don't normally do that. They just go for what the latest version is.

Right now, the version that we're using is the 2016.1 version of the records browser, though we are running on the 2016.2 accounts. But if you are going to-- the records browsers should be updated anytime soon. Try to check. So just change the version number on your records browser to verify if it has been updated.

So if it hasn't been updated yet, you can just still use the 2016.1. They don't normally change that [? many ?] information on the records browser anyway. So anyway, if you want to get information on a standard sublist, what you need to do is, again, go to the records browser, and then open the record definition.

So for example, if you want to get the attendee information-- or the attendee sublist is for your event record-- so what you need to do is you need to go to E-- or you need to click E, and then click on the event. So here, the example is the customer record. So you can click on C, and then click on Customer.

And once you click on the record, just scroll down to the sublist section. So the first section would be a set of fields. So this would be the body fields. So for example, for the message field in a task record, what you can do is you can click on T, click on Task, and then just try to look for a message.

So that's where you would get the information. But in terms of sublist, scroll down and then look for the section that says sublist. Now once you get to the section that says sublist, you would see the ID here beside the name of the sublist. So for example, the ID of the sublist here is attendee, but what you would see on the form is kind of covered by the line, but this should say attendees.

So the attendee sublist, the internal ID for that is not attendees, but attendee. And then below that, that's where you would see the internal ID of the sublist fields. Now to refer to the sublist or to reconcile the sublist field with the field that's displayed on the form, you can just look at the label. So here on the label, you can see attendance.

So if you look at attendance, it's almost always the same, but not always. So it's about, I think, 80% to 90% the same. For example here, attendance, if you look at the ID, it's also attendance. But if you look at the sent invitation to column, if you look at the internal ID, it's actually attendee.

So just be careful on that. Try to avoid guessing. Again, we have tools that are available for you to get the ID, so there's really no reason for you to guess the ID. So go to records browser, go to the field-level help so that you can get the information.

So let's talk about custom sublists. For custom sublists, this would be sublists that you attach to a record because those two records have an association. So what you need to do if you want to attach a sublist is that, first, you need to make sure you create a custom record.

Now so far, what we've done is that we've created a custom field which is a singular value that we are attaching to another record. But what if we don't need a singular value, like an employee code or coupon code? Maybe we need a set of values that we need to attach to another record.

For example, if you need training events-- so you want to know all of the training events that people has attended, or your employees has attended. You want to look at an employee record, and then just see a list of the events, the name of the training, when that was held, how long it was, or how much you've paid for that, something like that.

It's not just one single value. So it's multiple values. For that, you have to create a custom record. And then what you can do is that you can attach that record to another record based on a field.

So for example, here, let's say this is a customer record, and then you have your training event record here. What you need to do is that on your custom record, you need to have a field that represents your target or your parent record.

So if you want to associate your custom record to a parent record, if you want to associate that to a customer record, then you have to have a customer feel in your customer record. So you can think of that field as your foreign key. So if you're familiar with databases, where you want to perform a join, you would create a foreign key. And then you have an association with the key on one table, and then a foreign key on another table.

So something like that. Now if you're not too familiar with databases, you can just ignore that illustration, I guess. But that's one way of looking at it.

What we're going to do in this walkthrough is that we're going to associate a custom record, and then attach that to a standard record as a sublist. So what we want to do here is that we have a custom record called the performance review record, and we want to make sure that whenever we open an employee record, the performance review record that's attached to that employee record would be displayed.

So what we want to learn here is that we want to learn how we can create a custom record, and we also want to learn how we can associate a custom record to be a sublist on a parent record. So let's go ahead and use our performance review sublist so that we can link that as a custom child sublist to our employee record.

Now to create a custom record, what you can do is that you can go to Customization, Lists, Records, and Fields, go to Record Types, and then click on New. So again, that's Customization, Lists, Records, and Fields, Record Types, New.

Let me just go back to that option. And then after that, just define the custom record that you want to create. So for example, if you have a performance review record, you will just use performance review here, and then for the ID and so on.

Now our performance review has already been created, so there's no point in creating another one. But the idea of creating a custom record is simple. Just first create the initial configuration for your performance review, and then at the bottom, just go ahead and start adding fields.

And adding a field would be similar to adding a field to, say, a standard record for any of the transaction records. So anyway, let's go to the configuration of my performance review record. Let me see here.

So I have my performance review record. And then from here, you can see that I have several fields. I have a subordinate field that would store an employee record, because we want to know which subordinate this particular performance review record is for.

We want to know the review date, review type. We also have rating code, the salary increase amount. If this is a salary change review, the comments from the supervisor, and also the salary index. So let's see. Let me look at the records, and I'm going to hover over the More option here, and then click on View Records.

And it looks like we don't have any performance review records, so let's just create a few. Let me click on New Performance Review. And then here, let's say I'll create a monthly review. And this review would be for, say, James Rollings.

And for the date, doesn't really matter what date we're going to use. Review type-- let's say no salary change, rating code A. And yeah, that should be fine. Let's create another one. So I click on Save and Copy, but this time, I'm going to create a monthly review for another subordinate.

Let's say I have a review for Willheim Schueler. Let's click on Save. Pretty much the same settings here. And let's create a few more. So maybe I'll create a quarterly review for Willheim Schueler, and maybe change this to a salary change, and give Willheim Schueler a $500 salary increase amount. And that should be fine.

So we have three records. Let me go to my list. And as you can see here, I have two monthly reviews and one quarterly review. Two reviews for Willheim Schueler, and then one review for James Rollings.

Now if you have a custom record, you don't necessarily need to add that custom record to a standard record as a sublist. But that is a very convenient way for you to filter your information so that it's linked to a particular record. Now in this case, if I had, say, thousands upon thousands of reviews, it would be hard for me to look for specific reviews.

For example, if I'm looking for a review for James Rollings, and James Rollings probably, let's say, has worked for the company for 10 years, and has more than 20 different reviews. If I'm looking for a specific review, maybe I can just sort that by subordinate.

But if I sort that by subordinate, then my sorting for the name would be lost. And maybe if I'm looking for a certain salary review, salary change review, it's just going to be inconvenient. I'd still be able to find the record, but it would be harder for me to look for that if I have thousands of records to look that information from.

But if I link this to my employee record, what I can do is that I can open my employee record, and then inside my employee record, say if I open James Rolling' account-- so if I open this record, I can see that maybe just display all of the performance review records for James Rollings.

And if I do that, again, it will be easier for your users to grab the information that they need. And that's what we want to do. So to do that, what we're going to do is that we're going to go back to the configuration page of my performance review record. I believe that is this one. There you go.

And what I need is that I need a field that would allow me to link to my parent record or to my target record. So for example, in this case, I have a field called subordinate, so that subordinate field would store the subordinate where this performance review would be applicable for. And here, I can use that link to my employee record.

So if I want to link this to a custom record, I can't really do that because I don't have any fields that stores a custom record. Now if I create a new one-- for example, if I click on New Field and then choose, say, list our record here, and then choose Customer, now when I create this, I should be able to link this to my customer record.

So that's something that you have to remember. Again, if you want to link a custom record to a standard record, again, just have this field. And once you have that field, open that field configuration. Once you have that field configuration, put a checkmark on this option record-- this parent.

What this is going to do is that it's going to tell the system that, OK, this employee record-- whatever employee or whoever is selected in this subordinate field, I want that to be my parent record. And once I put a checkmark on that, inside our display subtab, a new option would appear, which is the parent subtab.

And as you can see here, if I uncheck that option, that option would disappear. Checkmark with that and that's back. And what that would do is that that would tell me, where do you want to store this particular record? And here, I can just choose which subtab would make sense for a performance review.

So for a performance review, I think it would make sense if I put that inside my human resources subtab, because that's related to reviews and performance. So human resources, I think, makes sense, so I'm going to do that. So parent subtab here, choose Human Resources, click on Save.

And now if I go to my employee record, and if I refresh my form, if I go back to my human resources subtab, if I scroll down here, I have a new sublist which is based on that performance review record. So if I click on that, you can see here that I have my monthly review.

And what's really neat here is that I can even create my own performance review record from here. So if I click a new performance review, that'll automatically create a performance review record for James Rollings. So if I want to create, say, a quarterly review-- I think we didn't really create a quarterly review for James Rollings. Let's choose any date. Let's say salary change, rating code A, and salary increase amount of, say, 800.

Save, and there you go. So if I go back to my human resources subtab in my performance review record, as you can see, I have two performance review records for James Rollings. And again, if I look at my list, go to View Records here, I should have four records now.

Two monthly reviews and then two quarterly reviews. So again, this is a very convenient way to associate records together so that you don't have to look for a record outside of your parent record. So you can think about this as a join. So you're joining two records together, and again, that makes it very convenient for you and for your users, and also as a developer, because that allows direct access to us from our script.

And then just a reminder, again, for the ID. And this is something that you will use. So a lot of people make the mistake of using the wrong ID. So remember, you need to use the field instead of the record definition.

So for the record the definition-- actually, in a sublist, you're not really going to use that. You're going to use, again, just your field ID. Now if you're going to get the value of a field-- and this is just for the sublist ID. If you're going to get the value of a field, just use the regular field name.

So for example, if I want to get my subordinate value-- so instead of accessing the whole sublist, I just wanted to access this one particular field-- then I would use custrecord\_sdr\_perf\_subordinate. And then these are the different entry points that we have on client-side. So we already know these entry points.

We were introduced to two additional entry points on the client-side, which is the lineInit event and also the validateLine event. The lineInit event is used to initialize a sublist, and then also the validateLine is used to, well, validate a sublist. So remember, initializing a sublist means that the user is actually creating a new line.

You can create a new line on sublist by clicking on any of the sublist buttons. So remember, just click on OK, Cancel, Insert, Delete, or again, any of the sublist buttons. With the validateLine, that is similar to your lineInit except instead of initializing a new line, it'll be triggered before going to the next line.

So for example, if you click on OK, Cancel-- well, not Cancel because that will remove the changes. If you click OK, Insert, or Add, then that would check the values that are on that particular line. So if you have several fields on that line that you want to validate, you can do that.

You can think of the validateLine, again, as similar to your save record, except that it's not going to validate all of the fields in the form, but just all of the fields in that particular line. So that'll be very useful if you want to do like a mini-validation.

What we're going to do this walkthrough is that we're going to do a few things. So first thing is that we want to count the number of existing performance review records, and then alert the user of that as we log into the page. So if the user has like five performance reviews, we're going to say, OK, display the number of-- like five performance reviews for this employee or something like that.

Next thing that we want to do is we also want to check if there are any performance reviews that has the rating code F. If they have an F rated review, then we want to inform the user of that. Also we want to default the review type of new performance reviews to salary change. So in case they're creating a new performance review, we want that value to be specifically set.

And we also want to set the salary increase amount so that it cannot be greater than 5,000. So we're going to add a validation to that. What do we want to learn here is that we want to learn how we can script a sublist, and also how we can use the lineInit and validate event on the client-side.

So let's go ahead and build our script. For this one, we're going to use our employee client-side script. Let me close that, and then let's open this. And by the way, let's change the version number here. And what we're going to do is that we're going to go to our pageInit script, and just change this to context.

And then on our pageInit script, let's create our employee record. So var employee-- well, not really create, but extract our employee record. So context.currentRecord. Now that we have that current record-- so the first requirement that we need is we need to count the number of sublist lines.

So for that-- so here, I'll call this the perfRev-- well, I'll just call this perfRevCount. Then I can get that from our employee record. So employee.getLineCount. So that's the function that we're going to use.

Now the getLineCount function or method has one property, or the payload object needs the property sublist ID. Now for the sublist ID, something that you have to remember is that you need to use the proper ID naming convention. So if you remember, our ID naming convention is recmach, which is short for "record machine."

And then we're going to get the field that we're using to link to our parent record. So remember, if I go to my configuration-- so for my performance review record here-- let me open that up-- common mistake is that people use this as they are sublist ID. Remember, it's not this. So this is not your sublist ID.

But you need to use the field that you're using to link their record, which in this case is subordinate. So I'm just going to copy that, go back to my code, and then paste that in. And that's pretty much it.

So the getLineCount is very simple. And let's go ahead and display this on the screen. So var alert, and then just say this employee has our perfRevCount performance reviews.

So let me modify this a little bit. We don't need var there. But anyway, let me upload the script, and let's see if this works. Let me go to my employee record here, and remember, for a client-side script, that would only work in-- whenever you're editing an existing record or creating a new record.

So let me edit this, and let's see if we get our popup. So as you can see here, I have a message saying this employee has minus 1 performance reviews. Now what does that mean? I have a minus 1 performance review?

Now whenever you see that message-- this has a minus 1 in your getLineCount-- that means one of two things. One is that you might have misspelled your ID. So that's the first thing that you would always want to check.

So go back to your definition-- so getLineCount. And what people do is that they kind of misspell this sometimes. Instead of using recmach with an "H," they spell the recmach without an "H." So make sure that that's correct. R-E-C-M-A-C-H, and then the field that you're using to link to your parent record.

Now it looks like my ID is correct, so it's probably not this one. So that's not the problem. Let me go back to my configuration. And the next thing that you want to check that, is your sublist accessible via the script? Now not all sublists are accessible via the script. So some sublists are accessible on the client-side, some sublists are accessible only on the server-side, but most are accessible in both.

So what you need to do-- one quick check that you can do is that you can go to the sublist, and here if I go to my performance review sublist, check if it's inline editable. If it's inline editable, then there's a good chance that that sublist is accessible via your client-side script. Now in this case, it's not inline editable, meaning I can't change the values directly from my parent record, and that's just because of a configuration that we need to change.

So we're going to go back to our custom record definition, and the option that we need to enable is this-- allow child to editing. So if I put a checkmark on that, click on Save, and let's wait for that configuration to save. If I go into my employer record, press Ctrl-F5 to refresh my form-- and let's see if we get the popup properly now.

So as you can see here, it says "this employee has two performance reviews." So now we know that our sublist works. So again, whenever you see a minus 1 value, that means that the script was not able to locate your sublist.

So one, check your ID if it's working. Two, verify if it's actually scriptable. So for standard records, there is an article in the Help Center. Just look for a scriptable sublists so that you can see all of the sublists that you can access via the script.

So anyway, that's our first requirement. So we were able to look at our performance review record. Now the next thing that we have here is that-- let me go to my-- where's that performance review record-- and let's change one of the reviews so that we can give this an F review.

So if you remember from our requirement, whenever we have an F performance review, we want the display that on our popup. So going back here to our performance review record-- so instead of having like multiple popups, maybe we can just combine that into one popup. So I'm going to change this alert statement so that instead of it being alert, I'll change this to notes.

So var notes is equal to that value. Let me change that. So that's my original notes value. And then we want to count the rating code. And if you want to count the rating code, that would be simple. Just create a for loop or any looping construct you're comfortable with, and then use that performance review count so that you can loop through the values.

So for my rating count-- so here, I'll create a variable for-- I'll just call this the fRatingCount, because we want to count the number of F rated reviews. So I'll just say that's 0, and then I'll add in a for loop.

So here, for loop. And I'll just use a for loop syntax here. So var i would be my counter. And then for my looping, I'm just going to use the performance review count. So anyway, my perfRevCount. My auto-complete doesn't want to work on me.

So anyway, here, perfRevCount. And then, of course, we want our counter to increase. i plus plus. And that's pretty much it. So here-- well, I want to get the sublist values. If the sublist value is F, then we want to increase our count.

To get the sublist value-- so here, I can say var ratingCode. So to get the sublist value, I'm going to use my employee record, and then I'm going to use the sublist value method. So if you're going to get the value of a standard field, you're going to use to getValue.

But if you're going to get the value of the sublist field, then that's going to be getSublistValue. Same thing with the get value. You also have get text. So if you're dealing with something that has a dropdown, then it's going to be getSublistText.

So for this one, the parameters that I need is-- first, you of course need to specify which sublist I want to script for, because you can have potentially multiple sublist values on your record. So in this case, I do have multiple sublist values, so I'm going to say that I want to get the value of my sublist ID, and then I want to get the field that I want to script for.

In this case, I want to get the rating code. So let me go back to my customization here, and then look for the rating code ID. If you want to access the field ID, you do not need to add recmach. You only need to add recmach if you are accessing the value of a sublist. So this would be your sublist ID.

But for field IDs, just stick with the regular ID. That's it. Now the next thing I need to do here is I want to get the current line number. Now for the current line number, since we want to look through all of the line numbers, then I'm going to use i as my counter. So here my ID-- or my line number, that would be i.

And yeah, that's pretty much that. So all I need to do now is just add in my if statement, and say if my rating code is equal to F, then I want to add that to my counter. So F rating code plus-equals to 1.

So we want to count the number. And if you are not familiar with JavaScript, plus-equals to 1 is just equal to F rating code is equal to F rating code plus 1. So this is just a shorthand that you can use in JavaScript.

Anyway, so I have that. And let's see. Yeah, that looks pretty much OK. So what we want to do now is just add into our notes to say notes is equal to this employee. Or maybe just add in a line break.

Let me go back to line 26 here. Add in a line break so that it moves onto the other line so that it's not one long line. So anyway, this employee has-- and then just [? to ?] count. Again, nothing fancy here. F rating code.

There you go. F rated reviews. There you go. And then of course, alert that to the system. Alert notes, and that's it. We're done.

So as you can see, whenever you're accessing a sublist, you're not limited to accessing just one line. You can access all of the lines if you want to. Just add a loop. And you can loop through that by using the perfRevCount count, so you know how many times you have to loop through your sublist lines.

So let me upload this, and let me go to my employer record. And let's edit this, and let's see if this works. And here, as you can see, it looks like I have-- I misspelled one of my rating codes. F rating count.

Yeah, there you go. Not F rating code, but F rating count. That's it. Rating code. Let me see if everything is properly spelled. Yep, that looks good.

Let me upload that again. And let me go to my employer record and refresh using Ctrl-F5. And let's see if this works. And there you go.

So as you can see, this employee has two performance reviews, and one of those reviews are F rated as we have expected. So that's something that you can do. So those are the two requirements that we have. Two down, and then two more to go.

So the next thing that we want to do is that if we scroll down here to our human resources subtab, and then to our performance review sublist, the next thing is that we want to default the value of our review type. So the default value for the review type is salary change.

So if I go to my review type, I have two values. So I have salary change and no salary change. Now these values are not records, but these are actually custom lists. You can create lists of values if you want to in the system.

So you can define-- so instead of you asking the user to type values, like salary change or no salary change, which they can misspell. And if they misspell that, that's going to be difficult if you're doing any kind of reporting. Because if they misspell salary change to something else, that would be considered as a different value.

So instead of having them type the value, you create your own list so that they would just select the value that you need. Now in this case, if you want to create your own sublist value or your own custom list value, you can go to Customization, Lists, List, Records, and Fields, Lists, and then New.

If I open a new subtab here, and then you can just add whatever values that you want. So for example, if I want to create a list of the different courses that my employees have attended-- so maybe I have like SuiteScript-- that's a course. I have SuiteTalk. SuiteTalk--

[LAUGHS]

SuiteCloud. What else do we have? Let's see. SuiteFlow-- you have something like that. So I can create my own list if I want to.

And with this list, I would have different values for my internal ID. So this would be text if I want to get the value of that. Or if you want to get the internal ID, I would have to use getSublistId.

Now for us, going back to our salary change-- the values-- we want to get the internal ID of your list, because we want this to be set to salary change. So going back to our list here-- so if I go to Customization, List, Records, and Fields, and Lists, instead of going to New, let me look for my review type.

So as you can see here, I have a review type field. And if I open my review type, going down here, I can see that, OK, salary change-- that's the value, but the internal ID is one. So if I want to set this to salary change, I'm going to set this to 1. Or if I want to set this to no salary change, I'm going to set that to 2.

Or what I can do is that instead of using set the value or set sublist value, I can use set sublist text. Now if that's the case, I need to make sure I spell that exactly as it is presented here. If it's capital "S," it has to be capital "S." If it's capital "C," it has to be capital "C."

If I misspell that in any way-- if I add in an extra space in between salary and change, or a space before salary, or after change, it would not work. And that is why it's preferred that you use the internal ID instead of using the value, because you can easily misspell this. So you can easily cause some problems with that.

Anyway, so going back here-- and then also-- oh, by the way, one more thing before we go to our script is that we want that to automatically set on a new line. The event that we want to trigger is called the lineInit event.

What the lineInit event does is that it initiates your line every time the user clicks on any of your sublist buttons. For example, if you click on Cancel, if you click on Add, if you click on Insert, if you are on the previous line and then you click on OK, or if you try to delete a line if you allow deletion on this option, then that would trigger your lineInit event. And that's what we're going to use for the script.

So going to our code here-- let me scroll down to lineInit. So I have my lineInit function here. Oh, by the way, let me make sure that I uncomment this so that we can use that. So going to my lineInit-- there you go.

So on my lineInit event, of course, as always, first thing that we want to do is we want to get a reference to the record that the user is currently manipulating. So I'll just rename this to context. I keep on renaming script context to context because I want it shorter.

You don't have to do that, but that's kind of a really nice shortcut. Just saves you some keystrokes, I guess, especially if you're using the context object a lot.

Now after that, we need to verify which sublist is the user manipulating currently. Now for this one, you can think of lineInit as similar to your field-level events. Because [INAUDIBLE] field-level events such as the field change, post sourcing, and also validateLine, you kind of have to filter to say, this is the field that I want you to monitor. And if the user does something to that field, then perform an action to that field.

Now in this case, what we're saying is that if the user performs an action on this sublist, then do something. Now in our case, we want to default a certain value. Again, similar to our field-level event, we have our if statement here. And then we're going to say context.sublistId. If that sublist ID is equal to our performance review sublist, which we should already have here-- I'll just copy that performance review sublist, and then put that inside here.

And by the way, what you can do is that you can define this as a separate variable. So maybe you can create an object here so that you can access kind of like your own mini-enum value. So that's something that you can also do.

So anyway, going back here, I have my performance review sublist. If the user is doing something to my performance review sublist, then I want to get my review type. And if my review type is empty, then go ahead and set that to salary change as a default value.

If it's not empty, then don't touch it because it's probably a value that has already been saved. So let's get the sublist value. So I have var, and then I have my review type.

Then for my review type, I'll have employee.getCurrentSublistValue. Now see here that I used a different function. So it's a slightly different function from the one that we used in our pageInit. So instead of using get sublist value, what I use is that the getCurrentSublistValue.

The difference between getSublistValue and getCurrentSublistValue is that the getCurrentSublistValue will move depending on where the user is at in your sublist. If the user is on the first line, then it'll automatically read that, OK, the line number that you're getting is 0. If you're on the third line, it'll automatically read that you're on line number 2. So that would adapt.

Now in our case here, on our pageInit script, we used to get sublist value because we don't want to wait for the user to move from one line to the other. In fact, we didn't really ask the user to do anything. We just read what's on the sublist and displayed that on the screen. Now in the lineInit function, here, we want the user to click on any of the buttons to initialize the line, because the user would be typing some values on the line anyway.

So anyway, for the sublist value-- again, sublist ID here-- for my sublist ID, that would be this value. We put that inside single quotes. And then after that, we need to get our field ID.

And then again, for that field ID, I can just go back to my definition and get my review type. Let's copy this. I do not need to add anything else to the ID. Just type that in, review type. And there you go.

And then after that, just add in an if statement to say if my review type is not empty-- so if my review type is not empty, then set the value. So employee.SetCurrentSublistValue.

So this is kind of similar to our getCurrentSublistValue, but instead we are setting the value. So parameter list would be the same, so I'll copy this. That makes it easier.

And then I also need to set the value property. So that means I want to set this to a specific value. And if you remember from our custom list, we know that one is salary change. Again, I don't have to use set value if I want to use set text.

I can-- "Text," not "test"-- but I have to make sure that I use salary change, and it has to be properly spelled. So it's kind of up to you. Most developers prefer using the internal ID because you are less prone for mistakes if you use the internal ID. But you do have an option of using the text methods.

There you go. So let's see if this works. Let me upload this to my account. Go to my employee record here and refresh this. So I'm pressing Ctrl-F5. And let's see if this works.

So I have those messages. If I scroll down to my human resources, and then go to a performance review-- and as you can see here, it's like-- OK, it looks like we have our script is not working, because the review type should default to salary change. But it looks like it's not defaulting to any value, so what's up with that?

Now remember here that your lineInit event would only be triggered if you are initializing a line. And the line would be initialized if you are pressing any of the sublist buttons. For us, we did not click on any button. We pressed Ctrl-F5, and that's the only thing that we did.

Now as you remember, Ctrl-F5, or loading the page, that would trigger not the lineInit event, but that would actually trigger your pageInit event. You're doing something to the page, not to the sublist.

Now if you want to do something to the sublist-- for example, the lineInit event-- and you can click on Cancel. And as you can see here-- ooh, I misspelled "employee." Let me go back to that and try that again.

That looks good. Well, let me refresh this, and if I click on any of the buttons on my sublist, that should initialize the sublist line and should trigger our lineInit. So I'm going to click on Cancel. And as you can see, after clicking on Cancel, my review type defaulted to salary change.

So as you can see, it works just fine if I do this. So let me click on Save, and that's it. So we were able to save the record. So three down, one more to go.

The first thing that we did was that we tried to display the number of performance reviews. We tried to look through the reviews, and then tried to get the value of our rating code. We defaulted our review type, and then lastly, what we need to do is we need to validate our line to make sure that the salary increase amount is not greater than 5,000.

Now the automation for this one is pretty much the same as your lineInit. So going down here to our list-- let me look for validateLine and then uncomment that. Then jump over to the validateLine definition.

validateLine-- there you go. Then on my validateLine, again, always just get that reference to your record. In this case, that's my employee record. I'll just change this to context.

So context.currentRecord. There you go. And now that I have my current record, similar to what I did on my lineInit, I'll just have a filter to say if my context.sublistID is equal to-- let me get that performance review record. There you go.

Copy that over there. So if the user is manipulating or doing something to my performance review record, I want you to first get the salary increaseAmount. So here, I have my increaseAmount.

Maybe I'll just call that-- yeah, maybe just increaseAmount. So I have my increaseAmount amount here. I can get that from my employee record. So same function. So getCurrentSublistValue.

The list of parameters are, again, pretty much the same. So I'll just copy those values, and put that right over here. There we go. Of course, I'm not getting the review type, but I'm getting the salary increase amount.

So let me go back to my record definition, look for my salary increase amount. So that looks good. I'll copy that, and then paste that right over here. Now that I've gotten my salary increaseAmount, the next thing that I need to do is, again, just add in my filter.

So if my increaseAmount [INAUDIBLE] there. There you go. If my increaseAmount is greater than 5,000-- so that's our validation. If it's greater than 5,000, then I want you to display an error message to say alert salary increase amount cannot be greater than 5,000. Just a simple message there.

Now this is also something that's a little bit different with the validateLine compared to the lineInit. Similar to your save record and also validate field, you need to make sure that you return a Boolean value. That is to prevent the user from saving that sublist line on your record.

So here, I'll just say return false if they abuse an invalid value. And at the bottom of my function definition here, I'll just say return true to allow changes if the values are OK. So let me upload that, and then go to my account here.

So I have my employee record. I'll edit this. And we should still have our old popup here. Scroll down to our human resources subtab, and then go to our performance review record. And then let's go to one of the existing ones.

So maybe I'll just change this value to maybe 6,000. So that's an incorrect value. Or 60,000 since it's already there. That's fine. So as soon as I click on OK-- so this would finalize my action.

So if I click on OK, or if I click on Insert, or any action that would finalize my line, that should trigger my validateLine. So the actions that the user would be doing on the lineInit and the validateLine event would be the same. So just click on OK.

And then after clicking on OK, you can see here that I have an error message saying that the salary increase amount can not be greater than 5,000. But if I change the value to something smaller-- let's say 300-- if I click on OK, as you can see, it just allowed me to save that line.

Even if I keep clicking on OK, it would not allow me to save that particular line unless I put in the right value. Now remember, validate field would prevent you from moving to another field unless you change the value. validateLine would not allow you to move to the next line or to another line unless you fix the field value.

Now you can use a validate field on your sublist. Now what will happen on the validate field is that instead of getting triggered when you click on any of the sublist buttons, the validate field would get triggered if you move from one sublist field to another sublist field. For example, if I use the validate field on my salary increase amount, if I move to supervisor comments, then it will automatically give me the error message.

I haven't moved to another sublist line. It's still the same sublist line, but if I want to add that extra validation, that's something I can do. So that is pretty much it. A quick review of what we did here.

On our pageInit event, we tried to count the number of sublist lines by using the getLineCount. We also looped through our sublist, and then tried to get the value using the getSublistValue. And then on our lineInit event, we try to default the value of your review type to salary change.

And something that you can also do is that you can add in a comment to say a one is actually salary change so that if somebody else would be looking at your script, they would know that-- what is this one value? Ah, so one is salary change. So that's the value of that.

Or I can even put in like one is equal to salary change. It really depends on you and how you want to handle that. But anyway, so on our lineInit, we used the getCurrentSublistValue. So if you don't know which field the user is manipulating, you can just use current sublist value. But if you do know which line number the user is at, then use getSublistValue instead of getCurrentSublistValue.

And then also for our validateLine event-- so here in our validateLine, it's kind of similar to your lineInit except you need to make sure you return a Boolean value. Return false if you want to prevent the user from saving the line, and return true if you want the user to-- or you want to allow the user. So that's it for this walkthrough.

Now let's move on to another section of our module. In this case, we're going to talk about scripting a sublist on the server-side. Scripting a sublist on the server-side is very similar to scripting a sublist on the client-side, but the only difference that you have is that on the client-side, you have your user doing actually doing something to the record.

For example, if they're on a sublist, they will be clicking on an empty line, and then typing some values in, then clicking on Add to finalize the changes to the record. On the server, that's not going to happen because they don't have direct access to the server. But what we're going to do is that since we're scripting on the server, we want to simulate the behavior that the user would be doing on the client-side.

For example, in this case, you can see here that I have my attendee sublist. So if I want to select a new line, meaning if I want to initiate that line or create a new line on a sublist, on the client-side, that would be clicking on Add or Insert, or just go to a blank line. On the server, I have to use the select new line statement.

So by using the select new line statement, that virtually creates a new line for me. I can set the value using the setCurrentSublistValue similar to what we've used in the previous walkthrough, or similar to a user typing values in on the sublist. And then if the user would be adding that line or committing that line, they can click on the Add button or OK if they're editing an existing line.

But on our server, we would be doing that virtually by using the commit line event-- or the commit line function. And that's what we're going to do. In this walkthrough, what we're going to do is that we're going to set up an event between the new employee and the supervisor.

So this would be kind of like a welcome meeting for your supervisor and your new employee to make sure that your supervisor would be welcoming that new employee to the team. What we want to learn here is that we want to review on how to find sublist IDs and field IDs. And also we want to look at how we can script-- or how we can use sublists on the server-side.

So before we start building our script, let's try to look at an event record. So I'll just go to Activities, Scheduling, Events, and New so that we can see how the event record looks like. And here, you can see that we have a few fields that are mandatory. So we have the title field.

If I look at that, it looks like the script ID is also title. We have the status, organizer, and also the event type, which are mandatory. But they're are already set with default values, though we don't need them.

And what we need here is that, we need to go to the attendees sublist. So in the attendees sublist, I can see that the send invitation to is mandatory, and also the response, but the response is pre-populated. So the only thing that I need to worry about here is the send invitation to field.

Now if I'm going to do this on a sublist, again, I'm going to click on an empty line, type in the value, and then click on Add. But since we want to do this on the server, we will be doing that virtually. Now regarding sublists, you need to make sure that you get the proper IDs.

If you're dealing with a standard record, you have to go to the SuiteScript records browser. Now with the SuiteScript records browser, like what I've mentioned, I would recommend that you bookmark this or you pin this to your browser tab. So as you can see for me, I have this pinned because this is something that I use a lot.

Now for my calendar event record-- so let me go to-- or not "C," but if I go to "E" for "event," and then I'll just click on the Event, you can see here that if I scroll down, I have a section here for sublists. And this would give you all of the scriptable sublists for this particular type.

So for my event record, I have my attendees sublist, my resources sublist, and also my time tracking sublist. Now we don't really need the resources and time tracking. What we're interested in would be my attendees sublist.

Now if you remember-- and if we take a look at my event record, I can see that the only thing I'm really interested in would be my send invitation to. So here, let me look for send invitation to. So the send invitation to field, the internal ID for that is attendee.

Now attendee here is also the same name of your sublist. So your sublist name is attendee, and your send invitation to field is also attendee. They're the same name, but they're referring to two different things. So you have to make sure you make that distinction once you start creating your code.

So let's go ahead and go to our IDE. So we're done with our client-side script. Let's open our user event script again, and we're going to use the same after submit script. And since this one also needs to be triggered when we are creating a new record, I will put that inside my if statement. So we already have that-- we might as well just reuse it.

So here at the bottom of that, I need to create my event record. So var event is equal to record.create. We already know this because we've used this previously.

We want to create our calendar event. So type, and then we have our record.Type, and then our event. Let's see. So I have my event record there.

Now after that, we're going to-- oh, by the way, since we are accessing a sublist, one thing that you have to remember is you have to remember to set the isDynamic option, and then set this to true. So this is something that is required if you want to script for a sublist, and the reason why-- by the way, isDynamic is there so that we can tell the server that whatever action I'm doing here on my script, that would be similar to what I'm doing on the server.

Because normally, if you don't set isDynamic, what happens is that all of the requests, it would be pulled into one operation, and thrown into the server as one. So if you're creating a new record, it's not going to do that line-by-line, but it's going to set that all at the same time. But with isDynamic set to true, that means that it would be evaluated one line after the other similar to what you would do on the client-side.

So on the client-side, you would be selecting the line first, setting the value, and then clicking on Add or OK, and then going to the next line. Selecting that line, adding the value, and then clicking OK or Add, and so on. So that's something that we also want to mimic in our server-side script.

Anyway, so that's that. So we have our isDynamic. Then after that, we want to set the mandatory field. So if you remember, event is a mandatory field-- not event. Sorry. The title is a mandatory field.

So setValue. I'll just set the title. And then for the title, I'll just use something simple and say "welcome meeting with supervisor." There you go. So after we have that supervisor, again, we need to select an empty line on my attendees sublist.

So event.selectNewLine. There you go. If you remember, this is what we saw in our discussion. Then for this one, I have my sublist ID, and this would refer to my attendees sublist.

So if you remember, the attendees sublist name is attendees. So it's not attendees, but it's actually attendee. So after that, set the sublist value. We already know how to use that.

So we have event.setCurrentSublistValue. So with my setCurrentSublistValue here, we want to set the sublist ID, so I'll just copy that. Not cut, but copy. Put that there.

And then we also want to set the field ID. And if you remember-- I mentioned this earlier-- the name of my sublist is the same name as my field. So attendee refers to my sublist ID. And the field-- that's the standard invitation to field. There you go. So the field is also attendee.

And then, of course, the value here would be my employee record for value. We already know that we can get the value by using the ID property. So the employer record here is the same employee record that we were logging information from earlier.

So going down here, after setting the value, of course, we want the user to click on the OK button or the Add button. Or virtually, we can do that by using the commitLine function. So event.commitLine.

And then for my commitLine, I have my sublist ID. And then for my sublist ID, that's going to be attendee. There you go. That's it. So that will be for my employee.

If I want to add the employee supervisor-- of course, this is a meeting, so it can't be just the employee meeting with himself or herself, but we also want to meet with the supervisor. So I'm going to duplicate this.

Select everything here, create a copy of that. But this time, instead of setting that to my employee, I need to set that to my Supervisor. So here I'm going to use getValue, then get the value of my supervisor. And that's it.

So just remember here-- and by the way, let me modify this format a little bit so that we don't have too many lines. We should be able to do this so that we can save a few lines here. I'll make it a little bit shorter. There you go. Change that.

And did we miss anything? I'll add this too. And let me change to this again. I'm just making it a little bit prettier so that it' easier to read. You don't need to do that, but that's something that you might want to consider doing.

So anyway, here, we've created our event record. After creating our event record, we've set the title. Then we selected our new line, set the sublist ID so that we can invite our employee, and then we commit that line.

And then here, we selected a new line, and then added our supervisor as an attendee, and then also saved it. But of course, I need to save the main record, because if I don't save the main record, then everything that I've done would be lost.

So here, I'll just do event.save. Remember, every time you use create, always, always make sure you save. Otherwise, everything that you've done would be lost. Now one behavior that you have to remember here-- going to my event record-- is that whoever the organizer is would be automatically added to my attendees list.

So as you can see, I am set as the organizer. And if I go to my send invitation to, then I have my name-- or at least Larry Nelson. I'm pretending to be Larry Nelson here. So that would be automatically invited.

So let's go ahead and try this. Let me see. Let me upload this first. I don't think I've uploaded that yet. So let's go ahead and test our script by creating a new record. So I'll go to Lists, Employees, Employees, New.

And then from here, let me see. So let's just create Test, 5, User. For my supervisor, I want to select-- let's say Baruch Rollins is my supervisor. And then for the subsidiary, again, doesn't really matter which values we use.

So let me click on Save. Let's see if this works. And it looks like we were able to successfully save our employee record, but let's verify if we were able to create an event record. Go to Activities, Scheduling, Events, New.

And then on my list here, as you can see, I have a welcome meeting with supervisor. So far so good. So if I open this, and if I look at my attendees sublist, you can see that Test 5 User-- my new employee was included in the meeting as well as the supervisor.

So for me, I was also included because I'm the organizer of this particular event. Now a quick review of what we did in the script is that this is normal create method. And what you can see here is that on the sublist, I use the selectNewLine so that I can select an empty line, and then I can set the value using the setCurrentSublistValue.

And here I've used the employee ID to include that employee to the meeting, and then persisted the line, or saved that line, by using the commitLine. Now one thing that you also have to remember if you're using multiple-- or if you're setting multiple line values is that you have to commit the line at every sublist change.

So for example, if I uncomment this line-- and that's what some people think-- like, I'm going to select a new line, and then change the value, and then select that new line again, thinking that it would automatically save the current line, and then move on to the next line. That's not going to happen, because selectNewLine would be similar to the user just clicking on an empty line.

So that means that if you've set the value for your attendee field, it'll override that value once it gets to here because you did not save your changes. So just make sure that if you are selecting multiple lines, you have to select each line independently. So that's pretty much it for this walkthrough.

There are some things to remember here. Make sure that you use the SuiteScript records browser if you want to get the IDs. Also check if a standard sublist is scriptable before you try to access it. Remember what happened when we were trying to access the performance review record, and we didn't it enable editing? So we kind of got an error because the script did not see the sublist.

So one of the guides that you can check is that if the sublist is inline editable, meaning the user can type in directly on the sublist, then that's fine. You should be able to [INAUDIBLE] script that on a server. Or if you want to look at a list of scriptable sublists, you can search for that in the Help Center.

So if you look for scriptable sublists or scriptable records, you would see which sublists are exposed to scripting. Also if you are going to use a custom sublist, do not forget to add recmach. And recmach-- and that's with an "H." I've seen some people put recmach without an "H." That will give you an error message.

Also you're going to add recmach only to your sublist ID, not your field ID. Then the ID that you're going to use should be the field that relates to the parent record that you want to add it to. Lastly, you have to remember that you need to use the related field as your sublist ID instead of the custom record ID.

Now it's your turn. Please complete the following activity. This concludes module 6, scripting sublists.