CRAIG CHAPPELL I think, for the sake of time, I'm going to go ahead and get started. There's apparently a line outside, and she's not letting them in until exactly 3 o'clock. But I want to maximize our time to give to you guys that are already in here. They might miss who I am. That's okay. So, let's go ahead and get started.

I really want to thank you guys for being here. This is Zero to Hero, Tackling Dynamo from Scratch for MEP Engineers. If you notice, any examples I give, I've actually turned off the 3D preview because I'm not using Dynamo to create any geometry, necessarily. I'm using it for engineering calculations.

This is the summary I have to show you. These are the objectives that I have to show you, that were in the handout, but what I want to reiterate on that is, I want to show you how I've done some Excel, using a Node to get a lot of information, trying to discover what I can use. Naming equipment by location in the model. So, I'm using lists with the geometry to figure out how to name them. And I'm doing a basic calculation to show how I can use sliders and other information with engineering calculations to get live input.

And then, I want to go back, after showing a couple examples, go back into some basic theories I think it's important to understand. And then, one I like is, if you have standards in your office for sizing components, you can create those in Dynamo, whether you want to do it automatically or not, and if you want to do it manually, you can show that. And then finally, I want to talk about how to make a peanut butter and jelly sandwich.

So, I'd ask that if you can please whole comments like about 45 minutes to speak, I've got some demonstrations I want to go through several times as I break it down further, and I also want to point out that with visual programming, there's a lot of different ways to skin a cat. So I understand that what I'm showing you may work how I did it, you may see a different way to do it, and that's fine. That's what I like about visual programming.

So, a little bit about me. My name is Craig Chappell, I'm the VDC Technology Specialist at TD Engineering-- TDIndustries, sorry. We do basically the lifecycle of a building, from engineering design assist, into virtual design and construction. We also have a group of guys that do building system analysis, to find efficiencies in the building, and of course, we do prefab and construction. I'm finishing up a four-year term with as an adjunct faculty at Kansas State
University, where I taught Revit to their fourth and fifth year seniors, and then I've had about 12 years experience in the AEC industry. Before that, I'm actually a proud graduate of the United States Merchant Marine Academy, and I have about 900 days at sea, on deep-sea ships.


So, how do you use Dynamo? Who has never opened Dynamo before coming to AU? Got a few. Who's done some playing around with it? Okay, and who's been using it in production? Okay, there's a few of you. Good.

So, what I'm going to do-- all these videos, I'll hopefully get to somehow, the PowerPoint, the notes, everything's for you, okay? So, I'm trying to have you pay attention so I don't lose anybody, because I'm trying to get a lot going.

When I first got into what can I do as an engineer and Dynamo, one of the things I realized is, you have levels. You have your categories, and then your elements, and your parameters. It's very important to understand those levels as you're getting data in and out of Revit. So, one of the things I like to do is figure out what can I get a hold of? So, I kind of want to-- this is a video-- again, you'll have access to it-- where I'm just showing you a simple demonstration of using Excel, pushing information to Excel. If you right-click on the graph, if you guys know, you can get the search bar.

So, all I'm doing here is I'm taking that Excel right to file, and I'm creating the nodes that are going to feed that. And I use this a lot because, how do I know what I can manipulate if I don't know what I can touch? So I'm going through here. And I like to use a select model element node, because I'll just go in and pick a water closet fixture, or any model, like an exhaust fan, an air terminal, anything you've got. So I'm just starting, and this is just a sample of what I was doing where I want to be able to see mass lists of information just to kind of feel what I'm doing.

And one of the things I like to show in here, in this little video, it's kind of important to arrange your nodes so your colleagues can see well, and what's nice, if you select them, and you right-click, you can get the aligned selection, and I can align the selection of nodes to the right, left, top, bottom, and I could also distribute them, or average them so it automatically pops them
together. And it's a great way to not play around with each node at a time. And that's it. What I would do at this point is just select a model element, run the graph and then in Excel, it would give me this list of every element that I can touch, so that I can start to learn what I'm dealing with and what I can adjust.

So, here's a question for you. Can anybody tell me, if you already have to have an Excel file created to run Dynamo, and not get any errors, anybody know? Who wants this? Who is it back there? Who is it? Look at that. I'm not liable for any injuries. False. What's really cool about Dynamo is that you share the script with a colleague, and they run it, and you don't have the file yet, it will create the folder, it will create the file for you.

Okay, so let's go into managing some calculations. Or this one where the Dynamo player, which is new, I have these exhaust fans in the model, and notice none of the marked values are there. When I hit Run, it's going to search, by level, what my exhaust fans are, and then number them from left to right. So, if you have a large hospital-- I used to do health care back in the day-- but a large hospital, and I didn't want to worry about naming them right away, I could run this Dynamo player graph, and it would do it. And what's really cool here, if I start to move them around, re-assign levels to show you that it's just playing with it, when I run it again, it'll update. So, it's looking through the model at the information when you run it, and updating it. Pretty cool.

So, that's the video of it, and here's the graph. I like to organize my graphs by color to help people understand what they're doing. So, I use the darker blue for inputs. So, I'm looking for mechanical equipment, exhaust, level, and one thing that's in the hand I mentioned, if you see these three-digit prefix, these kind of primers look kind of funny, my talk last year AU 2015, I went through this-- it's in a hand out, look for that video, on my web site there's information on that as well, I'm not going to talk about that here because it's not pertinent to this lecture.

So then, what I'm doing is I'm saying, how am I going to develop my selections? Because I'm asking Dynamo to look at every single mechanical equipment as a selection in the model, but I only want to see exhaust. So I'm reducing the selection down to exhaust, and then all of this is my action items, and I'm going to go through this again in a different set up of the graph, so you can kind of see-- right now I have the connectors turned off. The next example, I'll have the same graph rearranged with connectors.

So, I'm looking at the geometry. I'm going to number the values, and base by location and
level, and add the hyphen, and that way I can push the values back to Revit. So here's the
exact same graph rewritten, rearranged for you, and the examples that I'm going to give to
you guys is going to have both. One will be frozen so it doesn’t do it twice, but that way you
guys can see all this when you get back home. So, again here's my inputs, but in this case,
they are not necessarily in the same spot, and then here's where I'm developed this.

So, what I want to point out here is, here's my list of exhaust fans that I've been able to grab.
Notice that it's just a list of 10 right now. What I'm going to do here is, I'm going to sort them
based on the levels. If you look at the two left watch nodes, I have the list as it originally was.
Here's my level information. If you see here, it's in no particular order. I want to get it in order,
so I'm going to use my list sort-by key, and I'm going to use the list as my exhaust fan list, and
my key is my levels. So, what's going to happen here is on the output, is it's going to sort my
list of exhaust fans by the level, and I'm also going to get that list, my key, assorted keys, it's
going to give me that list here.

The other thing I do here is, this is great, this is from the rhythm package, so this is not out of
the box Dynamo, you will want to download the rhythm package for this. This is going to
remove prior to character, and this string right here actually has a space in it the character of a
space, because what's happening is right now is called level space one. So, I want to actually
extract the level naming from the model, I'm not using any string input, I'm getting this
information from the model. So, what I'm getting here is this assorted levels, I'm getting it
down to just the number. And if you get a project where the architect wants to play around with
a little bit, use some funky level names, you would just edit how you’re removing the text you
don't want.

Here's where it gets fun, because what I wanted to stress to you guys-- I didn't want to go into
Excel very much, because there's a lot of cool stuff in Excel, I could spend a whole other hour,
or two hours on Excel-- I wanted to focus on controlling lists. Because with engineering and
controlling calculations, I really feel like this is where the most important part of our time would
be. So, what I've done here is I've grouped by key.

So I'm using, again, I'm using my levels as my key, and if you notice, the exhaust fans, before
was just in one level list. I had all 10 in one single list. Now if I come back to this, notice that I
have a sub list of the two on level one, the three on level two, the two on level three, and so
on.
So, what gets fun is with the latest update of Dynamo, they added the levels in here, and I got to honest with you, I don't think I'm very good at Dynamo. I don't know why. But when I look at this and I think, OK, what is this [? et one, ?] et two, [? et three? ?] I had no idea what it was. And just with process of experimentation, I realized that once I split up my list into the sub lists now I can maintain this list and then sort the list in those sublists. So, what I'm doing here is by saying sort by key et level two, I'm keeping the levels in order, and as a secondary sort, I'm sorting with in those sub lists where they're located in the x-coordinate. You guys see that?

So, now that I have that I want to make a sequence, cause I need my marked values. I've got my level information, I need to get my dash here, the hyphen in the second. The next thing I'm going to do in this graphic, I need my individual mark numbers. So, I have two exhaust fans on level one, three exhaust fans on level two, and so on. So, in a sequence here, I'm using my count. How many fans do I have on each level? And I'm going to add those together in this list, and then I'm going to flatten it here, you see this flatten node, so what I have now is I have one, two one, two three, one, two, one, two, three. Because that's how many fans I have on each level in a sequence. All right?

And then finally I do a count of how many fans I have to get that middle character to get that separation, and what we end up with is, you see over here on the example, now I have a list for the exhaust fans in the order of their level and their x-coordinate geometry that gives me my values. And at pushes it back into the model.

What's nice about this graph and using the Dynamo player is I don't need to preset how many levels the building has. I have no I have no idea, it doesn't matter. It's going to look at the model, whether it has 10 levels or two levels, if there's a component that's not on a level, it just omits that. So, you don't have to come in and manually create anything. In the video, you notice all I did was hit the Dynamo player, and it analyzed everything for me and redid it. I like this example because, yes, I'm identifying the exhaust fans, but what other components might you want to model and add mark used to, and have it do it across your entire project in a second. Okay?

So, this is showing a little bit of some calculations. I kept it basic, I know it could get more complex with sizing a coil, and I'm going to go into detail about what it's doing. But I'm showing you the Revit schedule up here, and here's the actual view for the schedule, and this is the graph. And what I've done is incorporated basically my air across the coil, my heating water through a coil, what is my air as far as the CFMs the air. And that way I can kind of play
around with my coil size, my GPM requirement. And what's nice about this is in this example, I'm only showing three coils, but there's no limit to how many coils you could manipulate.

And what I did in this example is, if you see up here as I'm playing around with it, now I just switch from manual input of my heating water to a slider. I'm using an F node, and when it's untrue my slider becomes my input value. So I'm able to play around with my equations and see in real time what are my GPM requirements for those coils.

And what's really cool about that is if you're making Revit families, where your coils are pushing GPMs to your pipe sizes, you'll be able to see if your pipe system is sized properly by playing with it. In the kind of demonstration thought with this was, if we're working on a project and we just didn't get the performance from the boiler we needed, And we're only getting 110 degrees of air for the hot water, how would that impact our model? And we're not having to go through and redo the calculations several times, I can kind of play with it and see what we have.

So, this is the graph, again, set up in groups with the connectors turned off, so, that way you can see that I'm grouping by my inputs. Here's my selection, because what I'm doing again is I'm grabbing all the mechanical equipment, I'm reducing it down to what I want to deal with, and I go through the filter by building a mask in another example, to kind of explain that more.

Now, what's really important for engineering and Revit, is that Dynamo has no units. While you're calculating with N-Dynamo, all the numbers are unitless, but when you export the information from Revit into Dynamo, I like to call them absolute units, where length is in feet--for imperial, length is in feet, time is in seconds, and temperature's in Kelvin.

So, that gets really interesting, because if I want to do calculations with the information I'm used to putting in Excel or handwriting on paper, I need to convert them. And this is something I'll show you in a second, where I've created these nodes with predefined formulas so that while I'm doing these equations, I'm not having to write in what that conversion from Kelvin to Fahrenheit is every time. I can just copy this node for Kelvin to Fahrenheit. So when the information comes from Revit into Dynamo, I run through that conversion, now I'm good.

And what's important here, as you guys are getting used to this and trying to play around with it, you really want to have controlled experimentation. What is the unit coming out, what is it in Revit, and when I export the information into Dynamo, what does the watch node show me? And then try to figure out what unit of measurement that is, so you can figure out what those
conversions are.

And then what I'm doing here is-- these are the nodes I'll show you in a second too-- where I'm able to create predefined formulas, so that when I'm working in my graph, I'm not having to do custom nodes with all the equations in there. I have to find formulas somewhere else in the same file and I'm able to have just the inputs and the output. And this is a way to round to the nearest half of a GPM. So as the GPM goes up and down, I'm not getting tenths of a DPM, it rounds to the nearest half of a GPM for me. And then when it's all said and done, I push it back to the model, and I'm able to update all the information in real time.

So, let's go back through that one more time. So, here's all my parameter inputs. And this is where I'm getting my selection, I'm narrowing it down to what I want to deal with, all right? So, this is where I'm getting my air across the coil, my air flow across the coil, and what a lot of engineers-- you guys like to have safety factors in your design a little bit. Well, what this is here is that envelope loss, necessarily, where I can kind of add a fudge factor if I wanted to. If I wanted to make the calculation a little bit safer, this would allow me to add some heating requirements to that coil manually. And then here's my water in and out.

And you notice here, I'm converting each component, I'm converting it from cubic feet per second to cubic feet per minute. And then, I'm not really sure what the power absolute value is, but I was able to do some experimentation and use a conversion that works, so that's what that is. I'm sure I just don't know yet.

So, I'm developing my input from my calculations, and then once I have that, it's just a matter of adding the calculations where all the nodes fill in the inputs, and then push back to the model.

Okay, so hopefully that wasn't too overwhelming, because what I wanted to do was show you some neat stuff of what you can do. And now want to dive it back and do some basic theory and Dynamo. Is anybody lost completely? Am I losing anybody? Don't be afraid admit it.

So, this is a very fun example, just get something that works. And I actually like-- two times two was the one I was thinking of. So I'm going to start with the new graph. I'm using a string input, and just kind of watch it here. And actually, I've got some handouts I printed out if you guys want to pass them around, if anybody wants to look through them while we're talking. Look at them, passing around.
So, notice here that I'm using a preview node which can be helpful, because in this example, as I'm building it, notice that my first time I ran it that four a number. So, now I'm going to use a string from object. And what that's going to do is, it's going to take that number four as a number value, and convert it to four as a letter value, or as a string. And just to show you some changes, I changed that two to a three, when I run it again, I'm going to get nine as a result. But that's not a number, it's a string. It's really important to understand the inputs in Dynamo.

So, what I want to do next is, I want to take "my square is" as text, and then add the value I'm getting from my math. And notice that once you add a string or a connector to an already existing input, it detaches the old one and puts a new one in. So, when I run it again, now I have a list in my watch, and that shows me what I'm looking for. And just for example here, I'm going to drop down and lock the preview, so that as things change, I can see what I'm dealing with there.

And Concatenate is a nice thing where I can take the two items from this string, and bring them together as one string. And what's interesting here is, notice that my preview is still there, but my watch node zeroed out, it went back to nothing. So depending on how-- as you're building your graph, as you're starting to experiment and play with it, to me it depends on whether you want to use the preview of the node, or a watch node, because as you're changing connectors, if the run function is no longer done, your watch nodes will blank out. So in some cases, I like to go ahead and use of the preview.

And then what's also important, if you have a selection and you create a group, this is how you're able to group things together, as how you saw the colors before. I've seen a lot of presentations at AU where they're showing the Dynamo graphs, and they're showing some very clean organization of the groups. And then, you only get like 10 colors. I' a guy so I don't care. If you get like ten colors to deal with, I'm fine with that. So, again, I like to think of, from the spectrum there, you see the blue on the right, over to pink and purple on the left. So I think of blue through the warm colors of calculations, into pink and purple for results. That's just what I've been doing, it's not kind of any industry standard, but I do find it helpful.

So, that's basically-- once you have it, the groups, you can rename them. So that was just a kind of simple, I want to do something in Dynamo to have some success right away, play around with it. There's three categories of nodes in Dynamo. You have your create nodes, which have a little green plus sign, or you see the green bar. If you're looking in the menu
you'll see it's green there. Create notes are creating content. You have your query nodes, which are searching for content. You get a blue question mark or the blue bar. That's how you're asking for information from Revit. And then the other one is your action nodes. That's how you're going to perform or create an action, so you'll see a red lightning bolt or red line.

To use some quick examples, that range, or the sequence before that we use in a graph, I have a start, an, end and a STEP. Yes, you can use a custom code walk to do this very. Easily I'm just trying to show an example here. So that's a create node. A query node is like we did before, where I went to get the parameter value. So I'm looking at what element am I trying to get a hold of? And then what parameter I'm looking at, and that gives me a list of values. So it's looking for information from Revit to bring into Dynamo.

And then finally, my list by sort key, which I've really grown to love, that's an action. It's taking that list of information, and it's doing something with it. So, one thing I do want to go over for a second too, is it's really important to understand what are my inputs from my nodes, what are the nodes giving me as output? Because you can really get in trouble if you don't pay attention to this.

Billions are true false. I've got any auto-sizing, I want to go into that, a little more detail, but it's just a true-false. You can use this to control your graph, how information is going through it. And then it's important to understand the difference between a string and a number, because you can use numbers in strings, but in the handout I go through a demonstration of how, I think it was 3 plus 2.5 ends up being like 32.5 or something like that. Because it's taking the characters in the string, and just bring them together. It's not adding to numbers. So it's important to understand whether you're using strings or numbers.

And also geometry. Dynamo really started out for architectural and structural, I think for engineers, I think we're really starting to get a handle of it. I've really grown to like it a lot. But understand the geometry, elements have geometry that you can play around with. Lists, everything's a list in Dynamo. It's really important to understand how to manipulate those lists.

And then finally, you have your elements, and what I want to point out here is, when you have a list of elements, you're actually going to get your model ID number with it, that green highlighted model element ID. And what I learned today in another class, if I click on that, and it's in a view that's open, it'll actually snap to that view. So that's something I didn't know about until an hour ago. So that's pretty cool.
But what's important here is, if I'm dealing with a graph, and I lose track of my input, and I forget that I'm dealing with model components, notice here that in this watch node, I've got my model components. What's going on up here? I don't have any green numbers. So what's happening is my element get parameter by name is not looking at elements, it's looking at text. So I'm getting this error because back here at my filter, my list is coming from a text string output. If I do it the right way, I'm using my element's of category from here to my list.

Does everybody see this? Because this is what held me up for an hour or so on this. It's kind of like troubleshooting programming. When you think you're done with the graph, you think you've got Dynamo all ready to go, and you hit run, something's not right, now you got to go back and figure out, are all my connectors in the right spots?

So, what's happened here, and this is a mistake that I made, is I had to go through here and figure out where my input for my list filter is coming from and what happened was I was not using my model element list. So, I like to use watch nodes with almost every node as I'm building my graph, so I can see that, yes, I'm still using model components. I'm not just playing around with text.

So, here's a demonstration of how I was using my predefined formulas. And I found this through the Google search of Dynamo. I think the Dynamo Primer had an article on it, basically saying that you can take a custom node, I can define a formula, and then some basic code on what that formula is going to do.

And what I am doing here is I'm just double clicking on the graph to get a new code block, and then with these already in the graph, these have to be in the same file. I'm just going up here and I'm highlighting the definition here that has what it is and its inputs. And I'm pasting it into my code block, and then I'm also just grabbing the title up here.

This is a great way for your unit conversions, if you're going to have some equations in your graph, that you can centralize what the equations are going to do, so that if you have multiple instances where you're trying to do a lot of work, and you want to change or update an equation with a different idea, you're not having to go through and change six of them. You have a centralized spot where you can change it once.

So, who can tell me what is the subcategory of this node? I've got a little mean foam ball, stress ball, it's crying tears of joy. Anybody? Create, query, or action? I need a hand. Anybody
want to take a shot? Action. There you go. And what's nice here is, you start to think about it, the menus do show you here what you're dealing with. So this is a fun one. My background's in mechanical. I tend to, when I'm thinking of test projects or things that test to play around with, I tend to go to mechanical, so if you're electrical or plumbing, I'm sorry.

One thing I want to show here in this administration is, it's important to understand Boolean, and how to manipulate lists of Boolean values. So I'm creating a list here, notice that these values are purple, because when I'm creating that list, the value of true, the value of false, are definitions in Dynamo. So they're going to turn purple when I use them here.

This is a and b. So if a and b are true, it's true. If a is true and b is false, I'm going to get false. And then this other example here is if a or b is true, then the value is true. So what you can see here is I'm able to manipulate these. And this is important for a part of this next video I'm going to do.

And you can see here-- sorry, is this how I'm using this basically in a Boolean mask. So when I'm filtering the elements, what I want you guys to catch in this next demonstration on how I'm able to automatically control sizing of elements based on a range of another part, I'm using Boolean lists of true and false values, and I'm manipulating those lists. So, basically here's my list as an input. I'm using this Boolean mask, these are true/false as input and output. So this is a video where I've got some basic diffusers, I've got my types, my connection size, and my CFM. And pay attention to it as I change information. I'm not just changing the number, I know I just could cheat on that. I'm actually changing the connector size of that diffuser.

This was a fun discovery I had too. I had a guy ask me, hey, what happens if I change the size of diffuser? Dynamo can re-evaluate the CFM, and say, no, that's not a 6 inch or 8 inch, that's a 12 inch. And then, you're not always going to have the same type of diffuser, right? You're going to have a whole library of diffusers. So, I want to be able to manipulate one set different from another.

So, what's important here, is I have in this example a CD1a, and a CD2. Notice that as 300 CFM, it's OK on my two by two diffuser. But if I've got 300 CFM to a 12 by 12 diffuser, that's probably not likely. And just as a demonstration, I opted to make it default to 24 inches. Because you'd hope that if somebody is modeling this or designing this, and they see a 24 inch connection to a 12 by 12 diffuser, that's not right, right? I mean that's not good.
So, this is what we're dealing with. I'm getting my overall selection. And notice in this graph, the watch nodes, I've got all of my types from my demonstration. I've got 39 down here. But I only want to get, in this case, my CD1s. So here's my string contains. I'm using this node to evaluate that mark value of my diffusers. And what it's doing is it's giving this Boolean list of true, true, false, false, and so on.

The problem is, what if we have a standard where I have CD1s and CD1As. I don't want to-- if I have a list that contains CD1s, what's it going to do? It's going to grab all my CD1s and my CD1As. I don't want those CD1A's in this graph to continue on. So, I add another string contains, for that CD1A, but I include a not-A. So, this is a Boolean true-false. If you can kind of follow, they should go side by side here on this list. So if it's a CD1A, it's false.

So this is a list of the components in my model that are not CD1A. And if you can follow the logic, when I combine these two using this and, I'm getting a list of all my diffusers that are CD1, but not CD1A. Okay, so I'm able to reduce that down to one specific list. Once I have this list for my input, I'm going to get my CFMs, and this is highlighted blue, because in the next slide, this is a continuation. And I hope I don't lose anybody here because this is where it gets fun. Everybody having fun with this? Is anybody having fun? All right, this is great.

So, what this is doing is, this is actually conditional. I'm not getting any value out of these nodes except for true-false Booleans. And what I'm doing is I'm taking my CFM values from that list of my CD1A's, and I'm asking it, is that value of CFM below 100%, true or false? If it's true, it continues on. If it's false, I'm looking at it back down here.

Now, I don't want to start with, is my value below 1,000, because if it's true, it's going to continue on. You want to start with the lowest value, get those true values, and then like a if then conditional loop, as I go up in value, you're going to get your branches breaking apart. So, what I'm doing here is, this is my CFM ranges. I'm building my list of true-false Booleans. And this is really great because I'm actually going from the back, down. So, this is my if node. If it's below 100, it fills this first. So, here's my true value of, and this is in feet so that 6 inches is 0.5, half a foot. Right now, if it's false, then my input is here.

And this was a kind of mind-blowing, on tip of my head kind of idea, when I somehow figured this out, because I was just kind of playing around with an if statement, but then I'm like, what if I had two if statements that were together? And then I realized that by doing this, I could use my CFM ranges, and I could have conditional diameter sizing, and I could manipulate
everything with this. And it's completely adjustable. So once I had that list of Boolean that gave me my values based on here, then I can go back and set. So that's what that's doing.

One thing to point out, you don't have to make a list here, that was the early on, and was it was an issue that I didn't want to redo. Can you hold that thought please? Or do you have a question?

AUDIENCE: That [INAUDIBLE] condition, so is it possible put them off in that block code?

CRAIG CHAPPELL: You could, if you wanted to combine it. I'm trying to break it apart visually for you guys. Sure, I don't know. If you wanted to--

AUDIENCE: I'm asking you. I'm not sure. I'm asking you if it's possible.

CRAIG CHAPPELL: You can-- if you guys know, if you grab a whole selection, and right-click in that graph, you can convert it to raw code. In the handout, I do show a figure where I take this graph-- now that you asked about it, this was not planned, that was a good segue-- in the handout, I take this graph and convert what I can to that raw code to simplify it.

I found that when you're trying to convey this message of what you've created to colleagues, it's better to open it up some. Because I deal with a lot of senior engineers that are very smart, they don't really trust technology a whole lot yet, and they don't just want to take me from my word. So I like to document and over document what I do. So does that answer your question? I guess--

Okay. So, what are some of the things that you do? And while I was kind of playing around with strings, what I could do with them, I said, can make a sheet list? Basically just saying how many levels are in the building, how many times is that split, how many schedule sheets do I have, how many details do I have? This is my sheet list, blank. I cheered out loud when Dynamo Player came out, but I cursed it also because there was no input.

And believe it or not, about a week and a half ago, a gentleman from France posted a data shapes package, which gives you a Windows pop up box. So without that graph that was kind of intimidating, you never see it. And when I put in this information, I'm actually inputting that into my graph, and when I hit execute, there's my sheet list. It can be 2 levels, it can be 100 levels, you can have a split once, where there'd be no A or B. And while I'm at it, I'll just push it to Excel. And if you rerun it and change the inputs, it'll make a new sheet list. You guys like that?
So, how do you make a peanut butter and jelly sandwich? Has this been on the back of your head the whole time? Okay. This was a demonstration that I actually saw a long time ago, and it was, how do I convey a message with a good visual? You may think, okay, it's not very hard to make a peanut butter and jelly sandwich.

You walk into, let's say this demonstration, I'm not going to get here because it's going to get messy. You have a table, you've got a plate, you've got bread, you've got peanut butter, jelly, everything's there, and a knife. And you ask the group, can somebody tell me how to make a peanut butter and jelly sandwich? And they volunteer, not knowing what they're getting into, they come up on stage.

They're handed a mike, and they're sent behind the curtain, because then a second person comes up to the table with the instructions that they are to follow the first person's explanation explicitly. The first person behind the curtain has a microphone, they cannot see what's going on. The person in front of the room can only do exactly what they are told, and if you can imagine it very rarely goes well.

All right, so I use that as an idea to say, Dynamo, don't let it get frustrating. It can be easy. If you break down the steps. Take a goal. What do you want to do? When I thought about my sheet lists, how am I going to arrange my sheet lists? I'm not in Dynamo, I'm not in Revit. I'm just thinking. I'm writing a logic script kind of thing. How am I going to get my prefix? How am I going to get my levels? And you start to just piece those apart, and think of each piece separately.

And as I'm building these examples I showed you today, I got small successes that lead to larger successes, that when I actually figured it out, I'm like wow, this thing works. And then you're really shocked, and you get your coworkers over. They look at it like, "Oh, that's great." They look at these and they have no idea what you did.

So basically, you're not going to get into Dynamo first day and know what to do and to be honest with you, what really helped me, in my hand out, the last half of the handout are screenshots of all the nodes, or good examples of the nodes that are out of the box. I didn't have time to show the package nodes that I like to use, because frankly they change so much, and it's hard to keep up with that.

But what really helped me get beyond that beginner threshold was the sad truth of reading
nodes and getting an idea of what each node is available, and then having ideas of what can I do. And it's no way around it, if you really want to get good at this, you've got to put in some time. I probably spent a good two, two and a half days of my time, reading through the library of nodes, just getting in my short term memory what's available. And then reading what the inputs were looking, for what the outputs were giving me, so that as I was able to create these graphs, I'm getting an idea in my head. I'm starting to piece it together.

And there's no easy button for Dynamo. To me, the limit is what you're capable of doing. Anybody go the Dynamo slam last night, and see the guys take off? That was crazy, what those guys can do. And I'm like, I don't even know what to do with that.

So basically, I didn't want to name anybody on [INAUDIBLE] packages. There's a lot of great resources online. The actual Dynamo blog, The Primer, is kind of like your help file if you guys don't know about it. If If you go to DynamoBim.org, their web site, I believe there's actually a link to the primer. The forums is a really great place to ask a question and try to get help. I haven't really used it a lot, but I have read-- because sometimes when you Google a question, somebody has already asked it and gotten an answer.

And then what's really cool here, is the Dynamo dictionary is actually in beta. I caught wind of it about two weeks ago where they're trying to build a library of what the nodes are, what they do, as well as an example, a small little example of the input you need, the output you need. It's really cool, so please go check that out like I said, it's in beta.

But what I want to stress you guys is get on Twitter. There's a lot of developers of Dynamo on Twitter, posting stuff, and that's where I kind of got into it as well. A lot of what I've learned, I've learned through Twitter, that Windows dialogue pop up box from Data Shapes, a gentleman by the name of Mustafa from France,

I actually caught when he posted, that he posted on Twitter, that he sent this package to the Dynamo website. And I was really happy to get into work Monday morning. My wife couldn't understand it. But I went to work excited on Monday morning, because I was going to figure out that Windows pop up box.

And I'm not good with Python, but I have a little bit of history with computer science, background 2000. So it's been a while. But I was able to take what he posted, kind of figure out what he's doing, and tweak it to what I need. So, that's a lot of information here. There's a lot of packages to use, and I was wondering if anybody had any questions or comments. Yes,
AUDIENCE: How can you tell what's a no response, where its inputs and outputs?

CRAIG CHAPPELL: You'll get-- see if I can bring it up-- it's not very intuitive, but when you hover over a node, it'll give you a little pop out.

AUDIENCE: Yeah, I've seen them.

CRAIG CHAPPELL: You've seen them? It's the strings, the variables. What I like to do is when I'm looking at a node, I want to play with that one node and figure it out. The concatenate, where it bringing strings together, I thought I was going to take the lists and just punch the lists into that concatenate and get flat lists, with all the strings together. You actually want to transpose the lists, so, when you get that list, you want to transpose it, and then concatenate it. And I figured that out by just saying, how do I join lists into one string? And I wasn't worrying about how to change the world. I just wanted to figure that one little bit out.

So, to answer your question, I guess, is a lot of little experimentation, what's that node, and how do I get that node to where I can control the result? One more, yes, Sir? The question was, can you deal with Instance and type? Out of the box, it wants Instance. I prefer, I talked about it last year in AU 2015, my class. I prefer Instance-based, because I can manipulate as a type through schedules, so I have found that I avoid type. There are some nodes that will reference type-based parameters, but in general, I prefer Instance. So, in a way, the question's kind of yes and no. The answer, sorry. Yes?

AUDIENCE: You're doing that [INAUDIBLE] with the [INAUDIBLE], and CFM automatically [INAUDIBLE]. Do you have to run the Dynamo program every [INAUDIBLE]?

CRAIG CHAPPELL: Okay, the question was, with sizing in the CFMs and the connector, do you want to run that automatically or manual, right? In the example for the video, I have on automatic for demonstration purposes. As an engineer I tend to run manual, because if you keep it on automatic, that's going to be-- every time you make a change in Revit, that's going to want to update it and re-evaluate it. So, I tend to run my graphs in manual, and there's not much benefit to running that graph automatically anyway, because when you make some changes, you just run it again, and it updates.

So, I typically, even when I'm building a graph, I keep it in manual, because a great point I
heard earlier today was, never experiment with Dynamo in a live project because I routinely will crash Dynamo and Revit. And there's been a couple of times where that little spinning circle just kept going. I went to the break room, got a bottle of water, I went to the bathroom or whatever, and I come back and it's still going, and I had to force close the whole software, lost everything. Save often, experiment in a bogus model. Don't do anything with live production unless it's been vetted and ready to go. That's good. Yes, Sir?

AUDIENCE: The example [INAUDIBLE].

CRAIG CHAPPELL: Yes.

AUDIENCE: All these [INAUDIBLE] were out of the box?

CRAIG CHAPPELL: They were out of the box. What I did-- I don't know if it's a good idea or not, I've seen it both ways. Let me open it here. Where is my quill sizing? What I did was I renamed the header. So, it's doing the get parameter value, but I went ahead and renamed the parameter. I saw today earlier that a gentleman was using notes above it. So, this is actually the get parameter. You can see in the description what it is, but it's how you want to do it. I renamed the header to show that this is the parameter I'm getting from.

If you're curious about it, if you hover over it, you can actually get the description in the pop up. Some people have said don't change the node's name, and just add a note above it. But if you start playing around with your node movement, that note doesn't follow it. So I prefer to go ahead and rename it so that I can see that and move it around with it. Good? Okay. Anybody else? Yes, Sir.

AUDIENCE: Just slightly off-topic. [INAUDIBLE]

CRAIG CHAPPELL: Thank you. Yeah, follow the White Rabbit, take the red pill, thank you. It's the idea of, you're going off into Wonderland, you're taking that red pill, you're diving in the Matrix, because once you start to visualize this stuff-- I sit in meetings with project managers, and my notebook in the margins got, can Dynamo do this? Can Dynamo do that? Because they're talking to me about all this stuff that is going to take them hours to do. I'm thinking, okay, it may take me four hours to build the graph, but then it's a matter of seconds to run it.

So, I like to do Easter eggs in my videos to kind of see if you're paying attention or not, and I don't talk about it at all, so that's what that was, yeah. I was just having some fun. You guys
like to have fun at AU? Okay, you know it's not all about just numbers and crunching them. So, anybody else?

AUDIENCE: [INAUDIBLE].

CRAIG CHAPPELL: I'm sorry?

AUDIENCE: [INAUDIBLE]?

CRAIG CHAPPELL: Oh, yes. Everything that you've seen will be shared. The PowerPoint we shared if the video doesn't come through. The video should be posted, we're recording today, so the videos should be posted online. Go check out my class from last year where I talk about the evolution of a design model, about getting calculations into Revit models.

I've quit using formulas in my families because now I'm using formulas in Dynamo, because what happens when you send your families and your files to clients? They've got all your equations. What happens when you send families that are a shell because Dynamo is controlling everything when you send that to your client? They're getting the results, they're not getting all the engine. So it kind of keeps the IP in-house.

But yes, for you guys, I'm not hiding it, I'm not trying to hide anything back. If you guys want to reach out to me, you're welcome to. And actually, let me-- so, please, please, go on for this class. Give me some feedback, good or bad, I won't try to coerce you. But please go through all your classes and provide feedback. I believe there is a chance to win some stuff.

And also if you haven't seen it yet, when you go by Hall C on level two, there's an Idea Exchange where you can talk to them about feedback from the customer to that. And if there's nothing else, then I'll go and wrap this up. I really thank you guys for being here and being attentive, I guess. And thank you for your time.

[APPLAUSE]