MARCELLO SGAMBELLURI: Hello, everyone. How's everyone doing? All right! Everyone excited to learn about Dynamo? Yeah! Welcome, everyone, to the first class Autodesk University. We're also streaming live, so I'm very excited. Very excited that you're here. This is something that I'm very passionate about.

So, today we're going to be talking about more practical Dynamo and practical uses for Dynamo within Revit. So today, we're going to be talking about Dynamo and how it can help us in our everyday workflows in our office. So I want to get to know everybody.

If anyone's been to my presentations, they're really informal. I don't mind anyone asking questions or raising their hand and giving me comments. I love to hear all your comments. I love to hear what you have to say about this material, because we're all in this together. So I want to get to know everybody here.

So before I begin, let me just find-- get a poll here. How many people have actually used Dynamo, where they actually downloaded it and opened it and actually built some form of-- oh, excellent. OK, that's great. That's about 60%. Excellent.

That is not a requirement, though I do ask that everyone is somewhat familiar with Revit, because we won't be going over the Revit basics. But we will be talking about how Dynamo is going to interact Revit and hopefully save you hundreds of hours at the office. So this is what we're going to do. We're just going to go ahead and get right into it.

So I talked about this. This is what we're going to do-- give you gradual doses of Dynamo. Don't need any prior experience. And by the end of the day when you walk out of here, you'll know about some visual programming, if you don't know about it already. And what Dynamo is very good with is help you with those repetitive tasks in the office.

So I'm really excited about this class, because I've been preparing this material for an entire year-- been interviewing architects, things we do in our office, interviewing contractors, people in the industry, owners. I've probably sorted through about 100 different examples that I created. And I narrowed it down to about five that I'm going to show today.

But if there's something in here that, if you're needing more, there is a handout. And it does have additional exercises in the appendix, maybe 30 or so. And so I encourage everyone to go through that.

Now, how many people have actually-- it's not a test-- but how many people have actually downloaded the handout? How many people have downloaded the handout? Oh, excellent. How many people have looked at the handout?

Well, I always bring extra copies of the handout. So I'll pass these around. Make sure I get them back, although if I don't get them back, I'll be honored because I actually created something worth stealing. So do you mind, sir, to pass one to here and to here? There you go.

So I'm not going to do a huge intro, because I want to get right into it. We've got five awesome examples that I want to go through. Just remember, there's a lot of material out there through the handouts. And I'm very available. You can email me. My contacts are everywhere. I'll be here all week. I love to geek out about this stuff, so we can talk about it.

So if you're new to Dynamo, don't worry. As I go through the examples, you'll start to learn how to use it. This class was intended to be a part two to last year's part one that I taught called "Practical Dynamo." This one's a more practical Dynamo. Did anyone attend that class last year? About 20% of you. Excellent, good.

So let's just get right into it. I get excited about this stuff. Can everyone see-- can everyone see my floor plan views names over there? Maybe a little hard on the edge. Let me scoot it over, just so we're-- just so everyone's in line, because it is kind of over on the edge there.

Our first exercise is-- uh-oh. Let me see if I can get it over there. Oh, perhaps not. Anyway, you just got to look over there.

Our first exercise is something that we struggled with in our office and a lot of people have as well, which is your office standard has upper case text for views and sheet names and even text notes. How many people have a standard in their office where everything needs to be uppercase? And what do you do right now in Revit if you need to actually change these View Names right now? What do you have to do? Anyone? Go ahead. Anyone, raise your hand, anyone? You can shout out? What?

AUDIENCE: Dynamo.

MARCELLO SGAMBELLURI: Dynamo, yes Dynamo. No you're going away my punchline. [LAUGHTER] OK, I'll tell you what you do, right? You get the intern, go over here, you select on the View. You come over here to its properties, view whether it's a sheet or a plan view.

You come over here to the View Name. And you gingerly click on there. And then you manually type in what you just saw but in uppercase, right? Isn't that correct? There's no-- everyone on board with me here?

So I'm sorry. Was there a comment up there? So some of the stuff-- some of these examples just write themselves. What I'd encourage everyone to do is be a Dynamo hero. That's kind of my theme this year. I want you all to be heroes.

So when I learned Dynamo, I started saving hundreds of hours. And any time I'd visit our clients or architects-- I work for a structural office. And so we'd go to visit the architect, they'd say-- I talked to my architect friend when I was there. I said, are you using Dynamo? And his reply was, no, we don't use Dynamo, because all that computational design, we don't do it.

We do box retail. All our designs are square. Our buildings are square. We know the design. We know how it looks. We don't need to do Dynamo to do any of that for us.

And I said, what are you doing there? And he was working away over his computer. And he said, oh, that intern created all these sheet views and all these view names. And they're all in lowercase. I'm manually changing all these to uppercase.

And I said, Dynamo can do that for you. Let me show you. And I scooted him over. And I'm going to show exactly what I did. This is how it works.

So this is how ridiculously simple Dynamo is. Text to upper case, first we'll do view names. Then we'll use sheets. And then we'll do text notes. Watch this. This is so cool.

Ready? All you do is fire up your project and fire up Dynamo. It's in Add-Ins. And if you don't have Dynamo, go to dynamobim.org. It's completely free. It's an add-in. And it takes you about 30 seconds to download.

And I actually failed to mention who I am, didn't I? My name's Marcello Sgambelluri. I work for a structural firm, John A. Martin & Associates in downtown LA. So I got that out of the way.

Watch this. This is so cool. Now with Dynamo, it typically is better if you have dual monitors, because then you can work Dynamo in one monitor and then Revit in the other. But here, we only have one monitor.

By the way, how many people use dual monitors at work? Oh, OK. So we'll just have to shuffle these screens around. But anyway just bare with me.

Here we go. Ready? Dynamo. And isn't that counter supposed to be counting down? Someone? I see a counter there. I'll keep track here. They keep strict time because we're streaming over the internet.

So what we want to do with Dynamo, it's going to replicate the same process that you showed you what that intern does. So the first thing we need to do is select the View. Did you see what I did? But in this case, in Dynamo, you can actually select a whole bunch all at once. And in order to do that, we have to know a little bit more about Revit, which you all know, because you all said you're Revit users, right?

Take note that Floor Plan Views are actually system families. So if they're system families, there's a way we select it. When you open Dynamo and you want to select something in Revit, you go to the Revit tab over here. Then you come to Selection, pretty obvious.

And then you want to actually select the View Type. So you go under here under Element Types. Now, I know that's a little confusing. This actually, technically, is system family types. We're going to pull that list down, and we're going to look for the View Plan.

I'm going to come over here to the View Plan. There we go. Now, we just selected the type of the View Plan. That will take care of our View Plan. It's called View Plan, because that's an API language thing. It really should say floor plan, but anyway.

Dynamo-- you struggle with Dynamo with this language a lot of times, because Dynamo is heavily coded to interact with the Revit API. And it uses a lot of that same nomenclature. So you'll see things like that. But I'll help you along the way and interpret what that means. Just like Element Types, that's not-- it's really system family types.

So once we select the actual type, then we have to select every single one of those that's in the project. And all you do is come here and say, All Elements of Type. Now, what I'm doing is I'm dropping down these nodes, which are discrete functions. Now, when you want to pass information from one to the other, you pull information out of an output port-- just click on the wire-- and then you string it into on the left, which are called input ports. It's that easy.

Now, I do have to set something here called the Watch node, only in these instances where I select all these elements, because it doesn't have a good preview. It doesn't have a little watch button. But you can see now, Dynamo runs in lists and it actually select all the Plan Views. See that? It even gives the name, which is awesome. So now we selected all the Plan Views.

The next thing we need to do is-- do you remember when we clicked on the View and then we went over and we clicked on the View Name? So what we need to do then is we need to get the parameter of the View Name. Do you see that? So this is the second most powerful Dynamo node you're ever going to use in your entire life. And it's called the Get Parameter node. It's a beautiful thing.

All you do is go here to Revit and then, let's see, Element, and under Element, Element again. And then you see this Get Parameter by Value? Now, I'm going to use this set for later. So I'll go ahead and drop that down too. Let me scoot over, so everyone can see this a little better. Just keep an eye over here and what happens?

So now all we do is we say Get Parameter Value. Now, what we do is it's asking for the element, what we want to actually extract. And that's just right here-- all these Plan Views. And then all we need to do is we need to actually get-- it's saying, feed me a parameter name. And do you see how it says, it's asking for a string on the input port?

So what would that parameter name be? What would that parameter name be over there? Do you see it? What is it? View Name. So just type in View Name. It's case sensitive. If you have a hard time with that, that's just the way it is, so deal with it.

But what you do is, if you want to make a string-- we're going to talk a little bit about DesignScript-- you don't need to actually drop a string node down. There are string nodes in Revit. That's so-- using string nodes and number nodes-- I'll get to it a little later-- Dynamo moves so quickly that you always want to stay in front of the cutting edge.

So string nodes for Dyn-- I mean, string nodes are so summer 2015. We're not doing that anymore. What you can do is you can just Double Click on the canvas. And you get this code block. Anything you put in quotes will turn it into a string. This is DesignScript, but don't necessarily know that.

So we say View Name-- View space Name, be very careful how you write it. But once you get it, you got it. And then you feed it in like that. Boom! And then you pull down this little watch, and it tells you all the names-- [INAUDIBLE] the View Names. There's a little bug there where it's not reporting these, but it's just not showing it, but it's still pulling the value out.

So now, we actually got all the View Names. So now, just like the internet, the next thing you have to do is turn them all to uppercase. So now, we actually pulled all those View Names out and their strings. So all you have to do, Dynamo gave us a very special node called To Uppercase Case. Dynamo also has string functions, where you could do uppercase, lowercase, you can do searches, and you can add things, and it's really cool, a lot like Excel.

So in this case, we're going to go Upper. And if you can't navigate anything through the Dynamo window, just go ahead and type it in this search, and it'll show up as well. So you see this String to Upper? This is cool. We're going to go ahead and put that here and then feed that in, because we're feeding a string into the Upper. And then now, it just changed it to uppercase.

But do you notice it didn't change Revit yet? There's one more thing we have to do. We actually have to set the value to Dynamo. And that's the thing with Revit and Dynamo is-- I want everyone to wrap their head around the concept of how they work together.

We're working along, and we realize that it's very tedious to keep putting this uppercase, uppercase, and manually changing it. So then you look to Dynamo to help you. You fire up Dynamo. And you say, Dynamo, take over.

Dynamo grabs all these views, pulls out the strings, and turns them to upper case. And then it waits, because Revit can't do that-- turn to uppercase. And it's like, OK, what do you want me to do? Come on, I got this, what do you want me to do? And you're like, give it back to me.

And then Revit's like, give it back to me. So that's what we're going to do now. So we're just going to see this in and just update that Revit database. You got it? So Revit and Dynamo, they're just-- they're buddies. They work together and they create awesome things.

So watch this, all we have to do is take the Element-- it says Element. So anyone know what elements we're going to be pulling in here? These, right? The same elements, right? What would be this parameter name? Anyone? You can shout it out, because we're-- go ahead.

AUDIENCE: View Name.

MARCELLO SGAMBELLURI: View name, OK, cool. And guess what, we don't have to actually type View Name again, do we? We actually have one right here. See? So I can just pull out-- and you could pull multiple wires out of output ports, which is awesome. Thank you, Dynamo developers. That's really awesome. Boom.

Now, what am I going to use for Value? What am I going to use for Value? [LAUGHTER]

AUDIENCE: String to Upper.

MARCELLO SGAMBELLURI: String to Upper. Everyone, I know it's hard to see. OK, ready, watch the magic-- oh, I hope this works. Watch the magic happen. Oh, oh, oh. Did you see that? I got I got a note.

And now, Dynamo took over, and it said, hey, you want to change my View Names? Dynamo, you want to change my View Names? Well, do you want to name all the corresponding levels as well? Depending on your office, in our office, we say, yes. OK, ready, and it's done hopefully.

There we go. Do you see it? All uppercase. How awesome is that? Come on now, how awesome is that?

[APPLAUSE]

OK, cool. Now, it gets better. You're like, Marcello, you know what, that's all cool and fine. But you know what, oh, that intern, oh boy. You know what they did? They named all my sheet names, and they gave them all lowercase. Oh my goodness. What do I do? What do I do?

Well, sheet names are just the same as View Names, except they have a different parameter. Instead of View Name, they have Sheet Name. OK, cool. So ready? Here we go. So let's-- I'll get rid of that watch node. I will just copy everything. You can just copy and paste these.

Now, I want to disconnect this so I don't get in serious trouble here. Just, OK, so let's see. So now, Element Types, what do you think I'm going to be picking? Instead of View Plan, I'd pick View Sheet. Yeah, that's another API thing. I don't know why it's not called sheet view, but it's not.

So it's called Sheet- and you can type in-- let's see, View-- whoops-- View. And you can type here and it's smart, and then you can find the sheet. Here, View Sheet, View Sheet. Cool. And then this is not called View Name. It's actually called Sheet Name.

Now that should have pulled all the sheets. OK, cool. And then we're running it to uppercase. OK, cool. And then all we do is feed it right back in. You know what I want to do? I want to show you, when you click on a view, do you see here, it's called Sheet Name, right? So we got it right? Sheet Name? OK, cool. Just had to double check.

Ready. Let's feed this in. And now, we got our Sheet Views. How cool is that? All right. Awesome, right? No, come on now, how cool is that? Oh, you like View Names, not Sheet Names? This is more powerful, because sheets tend to be up in your face when views are kind of buried.

So now, we got View Names. We got Sheet Names. What's left, do you think? What else do we have that we can put to uppercase? Anyone? Text notes. Yes, question there.

Will that work for placeholder sheets? That's the question. The answer is, since I selected the Element Type, the System Family type, it'll work for any sheet. Does that answer your question? Cool.

So now text notes are a little funny. But there's nothing too challenging that we can't handle. So what I'm going to do is pop over here to my General Notes. So a lot of times, text notes are long, right? I mean, hundreds of text, hundreds of text, right? Pieces of text.

And so it works a little differently. But we still use the same logic. So if I were to actually select on the text node itself-- do you remember when we selected on the view, we could actually look at the View Name and see what text was in the parameter? If you select on a text note, there's nowhere in the instance parameters and nowhere in the type parameters that actually shows the text. So there's no-- you're like, OK, what parameter do I pull out, right?

Well, I don't know why Revit set it up this way, but the text in a text note, if you want to access it through the database, it's sitting in what we call Properties of the actual element itself. So Revit has a good set of out-of-the-box nodes that can get you about 80% there. But there's times like this when you need to create your own node to do something like this.

And so out of the kindness of my heart, I created a custom note that will actually extract the text for you. And I'll show you how that works. And everyone gets it, and the dataset is completely available for you to download. So don't worry, you get all this stuff. And it works very similar.

So let me show you how it works. So what we do is, in this case, we have to add in a particular DLL, like this. And it comes over here. Now, do remember when we used the get and set of the parameters? For a text note, I just make the text note a little analogous, so that you can get and set the actual text.

So I said, get the text note text and set the text note text. So that's what we're going to be using to actually make this to uppercase. So all we need to do is basically select all the text notes. So I will copy this again. You see this?

So this is Element Type. So instead of Element Type-- OK, everyone, what do you think I'd be pulling here? Come on now. What do you think I'd pull down that list, if we're going to select all the text notes? Text Notes, Text Notes, Text Elements, Yeah. Sometimes you don't know, and you just got to kind of look.

So Text, Text Note, Text Note-- oh there is is, on yeah, OK, there it is. Cool. So that pulled all the text notes. And let me get a watch note on here just to make sure I got it. Text Notes, there it is. Those are all the Text Notes.

Now, this says Get Text Note Text. This is a very powerful node, by the way. Everyone can have it. It's awesome. You pull that in there, and you pull that down. And do you see, it's actually pulling all the text from your Text Notes.

That's huge, because you can actually send this out to Excel. You can do other things with this very beautiful node. It's real powerful. And everyone can have it. Awesome. But there's some even better, which is the Set-- but let's get to that in a second.

So what are we going to do next, now that we pulled all the text out of the Text Note? What do we got to do now? What? Word upper uppercase. Yep, just go ahead and grab that again, Uppercase. Here it is, boom. Feed that in right there. Did it go to upper? Yep, Dynamo did its job. And then Revit said, please pass that back to me.

So then we can feed that into the-- see how it says New String Value? That's the actual text. And then let me zoom in for dramatic effect. And then our elements would be this one. And then you let go, and you watch the magic happen. Oh, now, how awesome is that, huh? Text Notes to Uppercase.

[APPLAUSE]

What would you have to do in the past? You'd have to copy that out to Word. You'd have to send it to uppercase and then send it back, right? So now, you save this-- it's a DYN file-- you save this as a-- you know what, why am I pointing? Let's do it. Let's see how it works.

Save this as a DYN file. So let's call it classupper, something like that. Now, you have this in your back pocket. It's also on the dataset if you don't want to rebuild it. You have this in your back pocket.

Now, all you have to do is go ahead and close everything, open up another project that you knew that intern was working on. And you go like this, Open-- one that you know for sure has text in lowercase, like this one, hopefully. Cool. Yep, here we go.

I guess this is similar, this is similar. And then we go ahead and we fire up Dynamo. And then we go ahead, and we open that classupper. I think that's what it was called, right? Boom. And then it says, corresponding-- oh yes, I do. And then you go boom. And then it propagates all those changes through your new project.

Come on now, how awesome is that? Isn't that cool?

[APPLAUSE]

Don't give this to the intern, because they're going to go golfing. They're going to do this and then go golfing. Don't give this to them. Be very careful who gets this. Oh, no, better yet, you take this, then you go golfing. [LAUGHTER]

So does everyone see how Dynamo can have a very practical application in your office? Just one example, we'll go through some more. And yes, sir. Yes. Oh, yes, this is a good question. OK, yes. And oh my gosh, oh, we're geeking out already.

Do you know what, do you have a spotlight? This is what happened. I don't have time to show it, but it's in the dataset.

When I created the text note-- and there's was a gentleman and I both worked on it at the office-- the Text Get and the Text Set, when I was like, I can get any text note, any value, no problem. But then I was like, wait a minute. I could then set the value-- I can set any text into a text note. How powerful is that? It's in the dataset.

But if you wanted to make live links through your general notes-- like, for example, so-and-so, so on and so on, see splice detail. Then all you have to do is just type out the View Name, right? And then you write a little script. It's this long. I already did it for you.

And then it'll pull the-- out of that view, it'll pull into the sheet data, it'll find out what number it is on your sheet. And it'll find out what sheet it's on. And then it'll just say-- it'll automatically populate and say, that's on one on S or one on A101. You see powerful that is?

So you can have smart text notes. It's a beautiful thing, I tell you. So I'm really glad you're thinking like that. I had that ah-ha realization, although I didn't think I was going to have time for it today. So I kind of explained it to you.

Any other questions about that? If there any others, I'll set up my next one, but I am listening to you. Anyone have anything else? No? OK, cool. Oh, another one? Yes, go ahead. I'm listening.

Oh, good question. It's a good question. Do you lose all formatting when you do that? The answer is, first of all, I'm-- first of all, the word formatting and the word text note don't really go hand in hand too well inside of Revit at the moment, at the moment. But what it's doing is it's just pulling the text. You're using the same element. It's putting it to uppercase and shoving that text right back in.

So it's not changing the text type. It's not changing the colors. It's not doing anything. It's just grabbing the text out and sticking it right back into the element. So it will not change any of your formatting. And it won't change any of the color or any of the type or anything like that.

I'm sorry. If you have some returns, hard returns, things like that, that's a good question. If you have hard returns or some special formatting, I suppose it could disrupt it. But just keep track of it when you try it. I haven't vetted that 100% out. So I'm going to say it's possibly, yeah.

Yes. Is it possible to do a text in a schedule? Yes, absolutely, although that would require another special node. But I can't write one for you right now. So let's talk later, and we can collaborate on it.

So let's go ahead and move onto something that's also near and dear to my heart. It is something that I have struggled with and something that's just simply awesome to talk about. Howe we doing on time? OK, good. We're only about 30 minutes in.

So here we go. Roof beams, roof beams, have to talk about roof beams. I've got a structural system here with roof beams. How many people model roof beams in their office? No one? A few? How many people know people who model roof beams?

[LAUGHTER]

Ah, that's better. So listen up, because you want to-- We have a lot architects in here? Come on, who are the architects? Cool. So what you do-- you learn this method, right? I'm going to teach it to you. And next time, you stroll into that structural engineer's office like this, and you go, I got-- I'm going to show you how to do those roof beams using Dynamo. It's awesome.

I'll run through it pretty quick. Right now, roof beams-- roof beams, when things are sloped, are a little unpredictable. Has anyone used beam system families? A few? Beam system families? Oh, but I just said, how many people use structural framing? OK, anyway. I understand. Cool.

Now, if you have some situations, beam system families work OK. Although what's really nice about beam system families, it'll put beams at certain placements along a particular path, along a particular curve or arc. But there's times when beam system families-- like if I wanted to put one on this bay, like this, they just sometimes-- they just sometimes-- they just sometimes let you down. You're like, oh, that's beam system fail. That doesn't do anything for me.

That's our new hashtag, beam system fail. Anyone tweeting? So I'm going to show you how to use Dynamo really quick. Structural beams basically are two-point click elements. So instead of using this-- I'm not going to say clunky, I'm going to say 16-year-old technology to model beams, we want to be cutting edge. We want to use 1-year-old technology, because we're cool.

So when you're a Revit user, you're cool. When you're a Dynamo user in the office, you're super cool. So you all are going to be-- when you go back to your office, you're going to be the cool kids in the office, I tell you. So you pass this, and you show it around. Even if you don't use this in the office, you want to share with people who do.

OK, anyway, watch this. So it's two clicks. So what you do is this. So what you can do is you can create a two-point adaptive family. It's really cool. So we go New, we go Family, and we come here to a Two-Point Adaptive.

How many people have used the Massing or Adaptive Component Editor yet? Anyone? OK, cool. So a Two-Point Adaptive, I'll go over it real quick. It's really simple to make a two-point adaptive with a line in between. And we'll use that as our guide to put the structural beam on.

Ready? This is how quick it is to model it. Ready? Everyone ready? I Who's timing me? Here we go. Ready, set, go. And we're done. It's that's simple.

So all you have to do is just basically place two points down, make the two points adaptive, and it's going to ask, where do you want point one, where do you want point two? It's going to put up a line in between. Do you see this right here?

I actually have one already set. What's a good idea is to-- actually, let's do that again. Go New, Structural Family. I do want to name it, so we go here to Adaptive Component, New, Two-Point, one, two. You can put them anywhere, doesn't matter. Select the two endpoints, Make Adaptive.

I do want to-- let's see. Oh, I can Load Into Project and Close. They're going to ask me to-- let's see. Is it going to ask me to name it? Wait, no, I want to Save As. Save As, we'll name it this, so ac W 16 by 26. We are using imperial today. Yeah, that's OK. And we want to Load it into the Project, and say Yes. So now we have it. So here it is.

So what's really nice about the Adaptive Component is you can place a two-point, and then you can actually structure it anywhere you want along the beam. So in this case, I have-- let me see I got to go to-- I guess I have to go to Wire Frame. Do you see here, I can actually tab on this end?

And then what's really nice, especially along the arc length, if you wanted to place this structural beam along the arc length, at like, say, the quarter point, all you have to do is type in here 0.25. And it'll actually get it right at the quarter point. Do you see that? And then the same with this side.

Anyway, that's not the cool part. So let's say you lay in your structural beams. So you're coming around here and you're rocking and rolling. This is what you want to do, lay your structural beam in. Boom, one there. Say you want the one there, whatever one there.

And then all you do is come over to Dynamo, watch this. Fire up Dynamo. Lay all your Adaptive Components in. Then you come over here, and then you could start a new Dynamo session. And then all you're going to do is build a structural beam. So under Revit, under Element, you go to Structural Framing, Beam by Curve.

Now, since this is 3D framing that's sloped, you can actually-- you're going to feed in a curve, which will be these in a moment. Then you can feed in the Level. Level actually could be any level. So you go here to Selection, and then Levels. It could be any level, so I'll just feed in level one. Let me get it over here a little bit.

And then under Structural Framing Type, that's the actual name. So there we go Structural Framing Type. And then this node will tell you every structural framing type that's in there. We're going to use a 16 by 26.

And then now the curves, this is what gets really cool. This is not necessarily to teach you how to do structural beams, but how to extract geometry out of Revit. I want to select every one of these Adaptive Components. So in order to do that, I'm going to go ahead and select Family Types.

Family Types are the loadable Family Types in the project. This has every single loadable Family in your project, which is really cool. So if you want to know what every loadable family is in your project, just drop down one of these Family Type nodes and pull it, and you see everything in there. It's really awesome.

So I'm going to select this one that we just created. I then do-- since I selected the Type, I have to select All Elements of Family Type, which is this one. Remember, we got to select all the instances of them. Now, we have them selected, but Dynamo doesn't know what a family is. It doesn't. Dynamo is like, what is this? So you have to convert it to Dynamo geometry.

So in order to do that, you have a node here under Revit, under Element. Then under element, you say Curve, and it says Element Curves. This tells Dynamo, reach into my family and pull out all the curves you see. Very powerful.

Doesn't have to be a curve. It could be a Solid, if you wanted a wall. It could be a Surface, if you had a curved surface. In this case, we're just doing curves. Boom. Look at that. Oh, that's so pretty. Look at all those curves.

Dynamo also gives you a preview of the actual elements. See, it looks similar, right? Those are the curves. And then all you do is just feed that right back in. Boom. And did you see how it built those structural beams right over here. Isn't that awesome? That's cool.

[APPLAUSE]

Yeah, no, that's really cool, because watch. Now what you do is you just tell Dynamo, OK, hey, Dynamo, just hang out for a while, let me do more business, let me get some business done.

You go over here. You grab your Adaptive Component, and you just start rocking and you just start rolling, like this.

And do you see any time you finish your transaction, it's building it right on the spot? Isn't that actual? Oh it gets better. Wait. Hold on a second, it's-- that's nothing. Watch this. Oh, that jumped on me, huh? That's OK. Leave that there.

Watch this. Because we made an Adaptive Component-- actually, I realized this about a month ago when I was prepping this stuff-- you can make it in-place mass and you can replicate the-- oh, let me cancel this-- you can replicate what we did with the beam. Let me see. Uh-oh, did I mess it up? Uh-oh, uh-oh, did I crash it? Uh-oh, I crashed it, didn't I? Oh, no, I lost my momentum.

[LAUGHTER]

Let's close the program. Dynamo is in alpha, so this is excusable, everyone. Let's close it, because I had some awesome momentum going in there. Let's see. Let's see if we can do it.

Let's get going again. Ready? Oh, we're doing great on time. Don't worry, everyone. Let's get up and running again. Any questions while I have fire this up again?

AUDIENCE: Mr. Sgambelluri, are there any log files in Dynamo. For example, if an intern in my office got a hold of it, is there any easy way to [INAUDIBLE]?

MARCELLO SGAMBELLURI: The clicks?

AUDIENCE: --to do it.

MARCELLO SGAMBELLURI: The question was, is there some kind of log file or tracking with Dynamo? Currently, not that I know, although it is possible if you were to code something like that, that is possible. So I can connect you, Will, with the developer to get you going on that, if you need to.

So anyway, let's do this. I got one midway. So I want to show you-- so we go Open. Let's see if we got one going on here. Get one final. So let's see, what family type do we have in here? We got the Two-Point.

So let's see, it's called Two-Point Adaptive. I'm just not going to recreate that name. That's what I'm doing here. Do we see it? There it is-- Two-Point Adaptive. Did it do it? Looks like we lost our-- so it looks like we lost our momentum there. But, OK, anyway, it's in the handout. I want to move on.

But all you have to do is place structural-- all you have to do is place Curves on it-- oh, never mind, I'm going to do it. I'm going to do it. No way, man, you're not getting away with this, Dynamo.

Here we go. Ready? In-place mass. Here we go. I just got to push Play again. OK, in-place mass, boom. Now, here we go, line through points, Follow Surface. All I'm doing is putting a mass line on top of this beam, which is also amazing, by the way. The Massing Editor is awesome.

I'm posting a point here and there. And it's actually going to follow that curve. When I do a class, I always like to give little tidbits in between, not just the core. So if you want to actually take a curve and make it follow a surface, whether it's a top of a beam or whatever, you can do Follow Surface, put two points down, or three, or whatever. And then you can over here, and it actually gives you relative location based on the beam.

So in here, I put 0, 0. And it would actually put it at that end. And this one would be based on the-- and this is UV-coordinates, so that would be probably four something, I suppose, Yeah, there we go. OK, we got that.

And let's just put one here on this end. And we're going to divide it. Uh-oh, here. Slightly off axis, yeah, but not really. And then we're going to divide the path, boom. So I guess we're at the 20% points.

And then all you have to do is come over here and place on one of your Adaptives that you just made. Click on this, and hit Repeat. And since this is an Adaptive Component nested within a mass, when you click Finish-- that's OK-- when you click Finish-- oh, I don't have it run automatically-- you go like that, then you see that?

How awesome is that, right? So you can actually take your Adaptives and replicate them. So take this and show all your friends. You'll be cool. Any other question before I move on? No? I'm sorry.

So you know what you do? This is what you do. I'm just-- OK, one more thing. Take matters into your own hands.

As an architect, you got a new opening or something. Put that framing in for the structural engineer. You'd be like, hey, you know what? I got an opening over there. Hey, structural engineer, can you frame it please, because I'm-- hey, how come you didn't frame that yet? Oh, you know, I got things to do, right?

So what you do is you go ahead and frame all that out for them. You just have yourself a framing party. And then you'll say, those changes, hey, look what I got for you, structural engineer. And they'd be like, oh, I love you, thank you. And they just need to verify it and size it, right? But it's awesome, it's just awesome.

So everyone ready to move onto act number three? OK, cool. Are we good on time? Oh, we got 50 minutes. Yes. Is that counting up or counting down? I can't tell right now when it's equal.

So I'm going to go out a little bit on a limb here and be a little risky. But I want to talk to you about the next thing, which is really cool. I've been developing this over the last year. And it's something that I just take near and dear to my heart, which is auto dimensioning.

Auto dimensioning right now inside of Dynamo is nonexistent. But I came up with the method to actually create dimensions using Dynamo and feed them into Revit. And I thought about some situations at which it would be applicable. And I'll show you some of those now. And later, if you want to talk about other situations, I'd be happy to talk about it.

But there's two that I came up with that I thought would be a good idea, just as an intro to show you how this node works. And everyone can have this-- the auto dimension note. Everyone ready? OK, here we go.

By the way, I always like to give you little goodies, and so this is actually a sample file. And it's got an actual telehandler in there, so you can have that, play with it, do whatever you want with it. It's completely parametric. I do teach a class on Thursday-- a lab, two labs on Thursday that will go over how to animate those kind of-- rotate those kind of things and move them around. So here we go.

So let's talk first about-- actually, let's go ahead and jump right into this slab. So here's our slab edge. And I thought, what we do in our office and a lot of our clients do is they have to dimension to edges slabs. Does anyone have to do that, dimension to edges slabs? A few of you? Does anyone know anyone who has to dimension to edges slabs? A few?

This will actually dimension to the edges slabs. But also, it does off-grid columns. Anyone have to dimension to off-grid columns? Anyone? Just a few? So it does both. So anyway, let's give it a try. Ready.

So what we do is we come over here and open up Dynamo. And then we come over here, and we can start with New. And then this is another custom node, so I'm going to go ahead and add it in Dimension. And it'll show up right here. And it's called Dimension Element. Everyone see that?

So this does two things. It'll actually do off-grid columns. So let me go ahead and copy one of these off-grid columns around like up here maybe. It does off-grid columns. Like, say you had some off-grid columns there that you actually need to string dimensions to.

Now, what this dimension node will do is it'll select all the elements that you put in here. And it'll auto dimension them to the nearest grid line in a horizontal and vertical direction. So watch this. So all we need to do is feed in the elements and the dimension type. So watch this.

So let's go ahead and select all those columns. I'm going to do this just one off with Selection-- so Revit Selections, Select Model Elements. So if you want to select elements in Revit, there's a lot of ways to do it. This is one way I'm going to do it. So I'll go ahead and select all these columns like that. I better run on Manual, just in case.

And then we're going to go and select the Dimension Types. Now, what I encourage everyone to do, if you're new to Dynamo, is go ahead and look through all the out-of-the-box nodes. And if there's something that you can't do, then head over to the Package Manager.

The Package Manager is an online resource that third party-- not, third party, but I'd say, private developers or just programmers have come up with custom nodes, like I'm doing today, who've posted them up on the internet, who you can then download. It's called the Package Manager. So I'll be using a few of those nodes as well.

And there's literally thousands of nodes out there that can extend your functionality within Dynamo. Go ahead and ask me that question. But I'm listening.

AUDIENCE: Can you save a selection set and pull that over?

MARCELLO SGAMBELLURI: Can you save a selection set and pull that over-- the question. The answer is, yes. Not with out-of-the-box nodes, but there is some of that in the Package Managers that I just explained.

So we're going to do the dimension. So first that I do-- so let's go ahead and pull the dimensions down. So we want to pull all the dimension styles out of the project. So this is a custom node that is in the Clockwork package. And it'll actually pull all your Dimension Types out.

Now, the way Dynamo works, like you saw these little green boxes, that doesn't mean a whole lot to us. But the way Dynamo likes to interact with Revit is it runs through its IDs. So if you see the ID, you know that it's been selected. And then in this particular case, we want to pick a linear dimension.

So then there's another customer out there called Dimension Type Style. We feed that in like this. And we can run that. Let's see. Here we go, Run. And then do you see, it's going to tell us the actual types of the styles.

So I want to pick this first one. Now what's extremely important inside of Revit-- in Dynamo, excuse me, is that you have a good grasp on lists. Lists are always-- list is what makes Dynamo powerful. And one way to actually-- so we want to actually extract this one out of the list. Do you see that here? We want to select number one, because that's the first linear style.

So in order to do that, if you want to select something out of a list, all you have to do is use the DesignScript again. So do you notice that this is based on a programming language. So 0 is the first one, and so 1 is the second one. And do you see how it's in those square brackets? So all we need do is replicate that.

So if you click on here, you can make a variable. Say x. And then you put square brackets down. And then you put 1, the actual index that you need. And you can pull that in and hit Run. And it'll actually pull that particular one out of the list-- just a cool little tidbit.

I'll feed that into the Dimension Type, and I'll feed that element in. And then I'll hit Run. And then hopefully, we'll see some magic happen. Here we go. Ready? Oh, they go. Isn't that cool? That's awesome, right?

Now, you may have four or five, but what if you got 500 or 600? Or what if-- this actually works for edges of slabs too. What if you got all those little nooks and crannies, you got all those little vertices. Some of the slabs we model literally have hundreds of vertices that these outer dimensions have to go to.

So think about that and figure out a way that you can actually use it. And if you can't use it in your office, then find someone who could. And be that Dynamo here that we're talking about. Yes. Could you use bi-level?

Yes-- the question was, could you just do this for level one if you had 10 levels? The answer is absolutely yes. It's just depending on-- it just depends on what's the active level at which you then run Dynamo on will then put the dimensions on that particular level and the codes in there as well. Cool. Everyone want move on?

Any other questions before I move on? Could you auto align dimension strings once you create it? You mean, auto align through Dynamo? Yes, but that's going to require some real complicated coding that would be way above-- OK, it's possible, that's all I'm going to say. But it's not in that custom node that I created for you. Yes, go ahead. I'm listening to you.

AUDIENCE: Can you have it auto take out the ones for [INAUDIBLE] as opposed to having just pull out the [INAUDIBLE]?

MARCELLO SGAMBELLURI: The question was, could you have them auto select? The answer is yes. All we did was just select them. But there's a whole pile of selection methods.

And if anyone wants to know more about selection methods, go to the dynamoprimer.com. And it'll go through a lot of the different selection methods, but I'm not going to go through them all. But the answer is yes, you certainly could do that. Yes.

AUDIENCE: What program are you using to create the DLLs?

MARCELLO SGAMBELLURI: What program are we creating to use the DLLs? To talk about custom nodes for a moment, there's three ways you can create. One, is you can just take all your out-of-the-box nodes and then you can group them together to make one custom node. Another way to do it is to-- and that's if you just want to use out-of-the-box. But you can condense it down to like one node.

Another way to do it is to use Python scripting, which is an internal language within Dynamo that accesses the Revit API. So you can use Python. And the other way to do it is you use C#, which is through a interface called Zero Touch. And these were created using Zero Touch. That's a little bit above, beyond this class, although if you want to talk about it, we can geek out about it a little bit more. And there's plenty of documentation out there on Zero Touch and how to make custom notes.

Cool. That is done. I am excited that we went through that. OK So let's go ahead and move on to-- actually, now we're moving on to act three. I'm really excited about this. I'm going to tell you what we're going to do, because I'm so geeked out about it, and we have time. Whoops. We got to get there.

Come on now. OK, went through that, went through that. Ah, here we are. Oh, topography. How many people love topography in Revit? Love it? Love it? Love it, hate it, love it, hate it, hate it! Come on now, shout it out. Hate it!

I'm going to make you love it. One thing-- OK, so let me back up a little. I taught a class two years ago here at Autodesk University on how to build hardscape and make it follow topography. It was big of interest of mine to figure out how elements could follow topography in Revit, because topography is a very difficult thing to work with. We all know that. It's difficult to build, and it's difficult to have elements follow it.

Are we kind of in agreement there? So I thought, well, if anyone wants to know how to model hardscape curves, roads, things like that, I encourage you to reference that class. It was called hardscape-- how to follow hardscape, how to follow topography, 2008, 2013.

One thing that always bothered me was property lines. How many people have property lines in their projects? How many people? A few? A few? How many people had to model fencing or any kind of like boundary element in Revit that would need to follow topography? Anyone, anyone?

So this is what I'm going to go over. I'm going to show you how to take property lines and fencing and then project them onto topography using Dynamo. And what's going to be better than that is, if you have any element that you want to follow topography, then you can use these methods.

So let's go ahead and just systematically walk through it. And this should take about the rest of the class. But if you have any questions, let me know. I am so passionate about this, because like I said, I taught a class of on it. And now, we have Dynamo as a tool to really help us out. So let's get going. Here we go.

So let's see, just let me get set up. Are there any other questions while I set up? Yes, sir.

AUDIENCE: [INAUDIBLE] use these on large projects? And if so, are you saying [INAUDIBLE] work sharing or [INAUDIBLE]?

MARCELLO SGAMBELLURI: The question was--

AUDIENCE: The major [INAUDIBLE].

MARCELLO SGAMBELLURI: The question was-- I'll just summarize. The question was, are there issues with work sharing when you're spreading Dynamo across multiple office-- either multiple offices or multiple users? There is a bit more of management involved with using Dynamo, although personally, everything I have experienced, it's not as hindered that-- it hasn't hindered that workflow. It's only made it better, if everyone on board knows what's going on.

And so for-- I mean, here's the thing with Dynamo. If you build a Dynamo script in Revit and you're running it to build all your things to uppercase, all your text to uppercase, say, you save it and close it. Someone opens up their local file across the office and starts changing all the text to lowercase. Well, how would they know about that Dynamo script? They don't, right?

But if you're not telling the office about the Dynamo script and what you're creating and how you're changing your files, then you've got bigger problems in your office. So it's important to communicate what you're doing. And once you build something like this and you hand it to someone else and they use it, they love it. So it's just a matter of communication and just management. But it certainly will enhance your workflow and save you hundreds of hours.

So property lines, here we go. So if you know with property lines right now, if you look at property lines right now, they're actually two dimensional, aren't they? Where are they? Where are they in 3D? They're nowhere, right? Just like rooms, where are they? They're nowhere. They're 2D elements.

So we're going to take them and make them 3D elements, and we're going to project them onto the topo surface. Now, what's really cool about this is, even if you don't work with topo and you don't use topo, how many people know people who need to work with topography and make elements follow topography? How many people have haters in their office?

How many people have landscape architects they work with that they just hate Revit, because they don't like working with this. Once I show you this, show them this method. And then get them on board to start thinking about how powerful Revit is and how powerful Dynamo is to help them with the workflows.

So let's do it. Realize that Revit topography is a mesh. It's a triangulated mesh. That's one reason why it drives us so crazy, doesn't it? Because it's triangulated mesh, actually. And it's the only mesh-- I hope I said this right-- it's the only mesh element that I know of inside of Revit. Someone could correct me if I'm wrong on that.

And so anyway, let's go ahead and work with it. But we can't work with mesh. We just can't. So let's go ahead and convert it to a surface. And Dynamo can do that. And let's go ahead and take polylines, those 2D polylines, and convert them to 3D lines. So you saw all that already.

So I'm going to move a little faster, because you're a little more involved with the process and what I've been doing. So I'm going to start throwing nodes down a little faster. Check the time, we got 34 minutes.

So the first thing we want to do, we want to take that topography. We want to select it. We want to turn it into a surface, so that we could then interact with it in Revit. So let's do we go. Here we go.

New, let's change to Automatic. Here we go. Let's select it, Revit, Selection, Select Model Element. Now, you could select System Type, I suppose, System Family. But we only have one big topo surface, so we're just going to select it like that.

And you know it's selected, because it has the element ID here. See, 10, 5, 5. If I were to toggle the eliminate ID in Revit, it would show the same number. Now, let's go and dig into the Package Manager. There's one node that is created within this spring nodes that will convert this from topo to a poly surface, which is what we need.

So let's go ahead and-- see, that's all we do. So we feed in the element in Topography, and we hit Run. This will chew on it for a minute only because it's just-- it's changing all of those meshes into a poly surface that we could then work with inside of Dynamo. Just these two nodes are extremely powerful in and of themselves, because it's going to create something that we can work with.

We can't just bring the mesh inside of Dynamo, because mesh inside of Dynamo doesn't allow us to do intersection, interaction. And it doesn't allow us to do any type of geometric interaction. That's why we're changing it to a surface, not a mesh.

Now, we're there. When I saw this for the first time, oh, my goodness, I was so geeked out. I was like, oh, that's probably the prettiest thing I've ever seen in Dynamo. Because before, you didn't have that. You had to work with this clunky mesh over there on the left. But now, we've got this pretty cool, sleek, new technology we're working with on the right.

So all we do here-- so cool, we got that. Now, let's grab the-- what do you think we're grabbing next? Come on now, what now?

AUDIENCE: Property lines.

MARCELLO SGAMBELLURI: Property lines, awesome. So we go here to Category. So you can select many different ways. Another way is the overarching category. And in this case, we want to pull down the Property Line. So we go, let's see here, Property, Property Line. Is it in here? No? Must be in the here in the Element Types.

OK, let's see. Here it is, Property Line. It's in the Element Types, not the Category. But we got to talk to those developers. They need to get it in the Category. So anyway, here we go. All Elements of Type, we pull that in. Let's grab a watch node on it.

And then we watch that. And all we're doing is pulling in the property lines into Dynamo, much like we did with the Adaptive Component lines. There it is-- Property Line. Now, Dynamo still doesn't know what a Property Line is. You're like, what is a property line? I don't know.

So do member what we did with the Adaptive Component? We told Dynamo reach into our family and pull out the Curve Elements of it. Do you remember that? So that's what we're going to do. So we're going to reach in-- Dynamo is going to reach in and then pull out the curves, and that's what we got. So now, that's the property line inside of Dynamo.

Now, all we need to do, in order to project it onto the surface, is we're just going to do an extrusion up and then run an intersection between that surface and that topo surface. And then we've got [CLICK] the Property Line. Now, there are some instances at which you may need to have an offset, because property offsets are just a thing of necessity. You don't usually build a fence right on a property line. You usually have some kind of offset, usually. You will.

So you can build in offsets as well. So what you can do to build an offset is you can take all these curves and turn them into a polycurve. So now, I want everyone to understand this. Do you see how we pulled out the geometry in Dynamo into Dynamo? Now, when we want to work with that geometry, we don't go back to the Revit tab, because we're not working with Revit geometry anymore. We use Dynamo Geometry tab, which is very robust.

See over here Geometry, there's a lot of things we can do in here. And one is to make polycurves out of those series of curves. Did you see how there were four of them? So we're going to do by Join Curves. All we're doing is making it one complete curve all the way around, just so we have the ability to offset it.

Now, we come up here to Curve. And then we can use an Offset Command, like this, like that. And then we can put in a Distance-- let's put in a Slider. You can use Sliders as well, if you're not 100% sure what it might be. And you can run that. Do you see that will create our offset. Do you see that?

Now, it's actually moving outboard, but we want it to move inboard. So we have to change-- negative would be inside, so this would be, say, maybe negative 40. We're using imperial units, and this would be 0 maybe. So anyway, so 0 would be right on the property line. What, 16 feet in would be 16 feet inboard. Everyone follow that?

Now, we need to what? Remember how I said, we need to take that and we need to extrude it up? So I'm just using all these cool geometry tools inside of Dynamo. Ready? So we come over here to Extrude, Direction, and Distance. See that? So we feed in the curves and we feed in the distance and direction.

Direction, luckily, is already pointing up, so we don't have to worry about that. The distance, we want it high enough to clear the topo. So I think maybe 100 feet or 111 feet would be good enough.

Now, we have two curves we want to extrude up. So in order to do that, we have to make a list out of those curves. So the way you make a list out of Dynamo is you can use a List Create node. But there's an even cooler way to do it, where we're using DesignScript. And it's called Use Curly Braces.

So anytime you have multiple elements and you want to create a list, just Double Click on your canvas. And then you come over here, you put in curly braces, and then you just put in two variables, say x comma y, whatever you want. Dynamo would know, OK, I'm going to make this a list. Feed me one list, feed me one element, and feed me another. And then I hit Run, and it's actually going to make a list of that. Do you see? Pretty cool, huh?

And now, I feed that right into my Curve Extrude. And then I hit Run. And it should extrude the curves up. Oh, yes, perfect. Do you see what I got now? Now, I got something I can work with. Everyone, bear with me now. We're going to keep moving.

Ready? The next thing we need to do-- what do you think, is what? Now that we got the surfaces, we got all the surfaces working, what do we need to do now to get that polyline all the way around the topo?

AUDIENCE: Intersection.

MARCELLO SGAMBELLURI: Intersect. This is the most powerful node we were ever given in the Geometry tab. I use this all the time. It's a beautiful thing. It's called Geometry Intersect. Oh, I tell you what, I got to tell you a little about my love for this node. I'm sorry.

Revit does not have a way to do intersections, does it? I taught a class in 2013 on how to model complex geometry, for anyone that wants to reference that. Half of that class was on how to trick Revit to do intersections. Well, guess what? Dynamo gave us one right here.

So when I first started learning Dynamo, the first thing I did was I typed in intersect, and I got that. And I was like, oh, the potential is huge. So all you to do is just feed in those extruded curves and then feed in that topo service. And then you've got that polyline. So that's a very-- that's a node that should be well-respected, because it has saved me hundreds of hours.

So let's go ahead and feed in-- it says Geometry and Entity, but anyway, it just means feed in two things. And the intersection of two surfaces equals what? Intersection of two surfaces equals-- a line or curve, right? And that's what we want.

So then we come over here to our poly surface, which is way over here. And do you see when you select on something in Dynamo, it will also highlight it. You feed that in. And then we hit Run. And then a very beautiful thing is going to happen.

Oh, yes. Now, you can also turn off the Preview. So I'm going to turn off the Preview of the surface, and we're going to get that. Oh, look at that pretty thing. Oh, my goodness. Come on now, look at that pretty thing.

[APPLAUSE]

And it even shows the preview in Revit. Look at that. Oh, my goodness. Now, you can stop there, because that's just super cool. Let me go ahead and-- but everything's in Dynamo, right? Nothing is in Revit.

So I'm going to show you how to take one of these lines and make a poly curve-- or excuse me-- an Element in Revit to actually follow that, because isn't that the point? Don't you want a 3D line in Revit. Isn't that what you want?

So what should we pick to model our lines in Revit, a good pick, our model lines? So all we're going to do is extract out this exterior curve and then make it a poly-- excuse me-- and tell it to create model lines inside of Revit. And it will put it right on that property line. And then we'll get to the fencing.

So the way you do that is we actually need to extract out-- so we got the Geometry Intersect. Let's all realize that we have the Geometry Intersect. Now, it made these little lines, but that's OK.

So now, the first one, if I collapse this list a bit, this first one right here is the actual outside. And how do you know that? There's a few ways. Let's go ahead and use our List Extraction. Do you remember our List Extraction method? You just say x, and let's say we want to pull out zero. And then we feed that in there.

But do you notice how it's a list of a list? So do you know how you pull out lists of lists? You just follow that same format. You see how it says 0, 0. So then just do 0 and then another 0. So easy. Oh my goodness, DesignScript is beautiful.

And let's hit Run. And then it will then pull out that list. And when I select on this, do you see how it's selecting that outside line? It's a little hard to see, but it's selecting the outside line. Let's hold onto this. We'll send that out to Model Lines. No, let's do it now.

So now, if we want to build Model Lines inside of Revit-- if we want to build Model Lines inside of Revit, what tabs am I going to go to?

AUDIENCE: Revit tab.

MARCELLO SGAMBELLURI: What?

AUDIENCE: Revit tab.

MARCELLO SGAMBELLURI: Revit, yes. It's because we're now building Revit geometry, right? So this, remember, is the outside property line projected onto the topo. It's not the offset line. So we want to go to Revit. And under Revit, you go to Elements. And there's a lot of nice elements you can create here, and one is the Model Curve. And then we do By Curve.

Now, we already have the curves here, remember. Curve is a general term for a linear element, whether it's a straight line or a curve. So I'm going to feed that in. And now, when I hit Run, it should create that model line right there. So let's go ahead and see what happens.

There's a lot of them, and there are little piece-- Oh, there. Oh, that's pretty.

Did you see that? Oh my goodness. Wait, where'd it go? OK, it's there. Oh, now, let me select it. It does show the Preview, and we can turn off the Preview. I don't necessarily want to do that. Anyway, it's there. You saw it?

So next is we want to deal with this intersect. Now, does anyone remember what this intersect is? This intersect is this series of lines that are on the inside. Those series of lines on the inside are the offset.

So I'm going to show you how to build fencing around the inside. We'll just do posts, but we can build a wall if you wanted to, but in this case, I'll just do a post. That's that list right there with those series of lines.

So in order to do that, I want to divide that up into a series of points and then put posts all along there. So what I want to do is actually turn that back into a polycurve, because it's all these little lines. So what I do is go use my Polycurve command.

So we go back to Geometry again. We go to Polycurve. And then we go By Join Curves. And now, I have to extract this list too. So in order to do, that I go x. And then now do you see-- oops, that's curly braces. That's going to mess us up.

It says 1, and then 1. So we go 1, and then one more is 0. We pick the first one. By the way, you do these code blocks like this in your office, I mean you're just going to blow people away, because this is just like the coolest thing in the world.

The way you had to do this way back in the summertime of 2015 is you had to actually use these get item by index. And you had to kind of nest them together. And it was a mess. So now, it's really beautiful.

So anyway, we go like that. And we hit Run. And it should pull out that next one. Yes, cool. It pulled it. I'm just going to click on it. You can see, it's kind of highlighting there. Cool.

Now, I'm going to pull that into the polycurve. Let's keep going. Now, this will create the polycurve. There we go. Oh, yeah. I'm selecting the Polycurve. Now that I have the polycurve, I need divide it up into points.

Now, this is up to you how far you want those fence posts. But let me show you some tools in Revit that are really cool. The one thing you can do if you have curves in Revit, go over to the Curve tool, is you can find its length. It's really awesome. Here, where's Length? Here is Length.

This is a long fence. Oh my goodness. What is that, 2,000 feet? Holy smokes. You can find its length, right? So what we want to do is we want to place fence posts at, what, 12 feet maybe? I mean, that's maybe a good guess. Or you could make it tighter. I'm going to show you how to divide a-- You got to put points along a curve at 12-feet increments.

First thing you need to do is learn the length, which is easy to do. So you got the length. Next thing you need to do is put points all the way around it. Now, there's something in here called Point at Distance. This is a very powerful node. We're putting points in Dynamo around the curve.

So we feed in the polycurve. And then the distance-- we could put zero, and then it would just put one point at the start. But we don't want that. We want one at 0, 12, 24, 36, so on, all the way to 2,000, whatever.

So in order to do that, we use something called a range-- you create a range list. And it's really easy. All you do is Double Click here again. This is what the cool kids do. And you create a range. You list where you want to start, the number range. You want to start at zero.

Then you give it two periods. And then you want to tell it where you want it to end. In this case, we want to make a variable called Length. Now, let's see, we good there? Cool, Length.

And then we want to give it two more periods. And then you want to tell it the step amount. Simple. In this case, I'll just say 12. Cool. So all I need to do is feed this into Length, and it'll know that. I'll hit Run, and we'll just verify that we got all of our numbers correct.

Here we go. There it is. See? 0, 12, 24, 36, got it. See all that? Aw, cool, that's awesome. Now all we do is feed this in here to Point at Distance. And then we hit Run. And then it should put points all the way around. Yeah, there we go.

And do you see, it actually created-- how many points? Let's see. It created 191 points, although we could have tightened that up. And you can make it 1,000 points, 2,000 points. And does anyone know the limitation on an array, if you were to try to do this using Revit? What's the limitation on an array? Does anyone know that value on array?

How about the limitation on a divided curve in Revit? Anyone know?

AUDIENCE: 2,000?

MARCELLO SGAMBELLURI: No, it's 200. So if you wanted 5,000 fence posts around here, Dynamo exceeds the limit at which you can divide a curve using Revit. Do remember how we divided that curve into six, so we could put those that Adaptive Components on there? If I tried to type in 10,000, it would say, oh, you can only make 200 divisions. Here, you make an unlimited amount.

So all we're going to do then is place fence posts on here. Now that we got the points, we're going to place families in Revit that'll represent the fence posts. And then we will be-- oh, my gosh, we are doing awesome on time. So what's nice is it shows the preview, right?

Now, this is something that everyone's going to use in their lifetime. It's a very beautiful thing. Once you have points set up in Dynamo, you can tell Dynamo place the family in Revit, any family I want in Revit, any loadable family I want in Revit at these points. It's huge. I mean, just think of the impacts.

I was talking to the gentleman over there who needed to set, what, kitchen equipment, things like that. So its applications are just endless. So now that we've got all the points, cool, let's go ahead and drop down another one of my favorite nodes, which is Revit, because we're building Revit geometry. We're not building it, but we're placing it. And then under Elements, under Family Instance, we have By Point.

So now, this will place any family we want. So I'll feed in the points. And then it's asking for a Family Type. Now, for a fence post, I already went ahead and made you one. It's called wire fencing.

It's just a cylinder, but this could be anything you want. It could be a picket fence. It could be refrigerators, I don't know, whatever you want. You could have some fun with it.

Oh, you know that articulating telehandler that I gave you in the other example, put that all the way around and see what happens. That would be fun. Just have fun with it.

So then we ought to pick the Family Type again. So there we go, Selection, Family Type, and then it was called wire fencing, right? Wire fencing, here we are, wire fencing. You feed that into our family node. And then you hit Run. Boom. And then you watch some awesomeness happen, hopefully.

Yeah, look at that. Oh, isn't that awesome, fence posts? Cool. OK, cool. Let's Save this. I'll just save it again, OK, whatever. I'll Save it.

Now, here's the cool thing, right? Remember what's happening here. Wow, we built a nice one. That's a nice one. By the way, that's probably about 18 nodes. I usually don't build Dynamo graphs more than 15. This is kind of an exception, just because if you're doing simple, everyday things, the nodes don't need to be that big.

And don't get intimidated. Sometimes, you open up those graphs, and they just look like fuzz, because you have thousands of boxes and thousands of wires. Don't get intimidated. Just make your code really tight.

Most of them, we had just four or five nodes. This is, like I said, about 18. But they should be around 15. Once you get into 20, 25, start thinking, am I really be efficient here? OK

So anyway, now, here's the cool thing. Watch this. Because-- oh my gosh-- I get so excited about this. Because Dynamo is constantly tracking-- well, we're in manual mode, but anyway. Since Dynamo is constantly tracking this topo shape, it's constantly tracking where the property line is, it's constantly tracking where the intersection is, it's constantly tracking all the divisions and all the fence posts that it placed on there, this could represent start grade.

But you know how it is in Revit and in your projects. That topo surface changes. You get new site data. You get new elevations. Things just change, because topo is organic. I mean, a lot of times, they don't know where final grade is.

You set something like this up And, then you were to actually edit your topo surface-- say this is your start, and then it would be something like maybe a little simpler, like I don't know. Let's just wipe out all these points. And then this would now be our final grade. If I were to hit OK and I were to run this again, then everything will automatically update. How awesome is that? Now, that's cool, right?

[APPLAUSE]

OK, cool. Now, it's got the Preview on there, so you can't really see it. But let me show you. I will close Dynamo now. Thank you, Dynamo. You're awesome.

And this is your final product. See it? How cool is that? You got a 3D topo line. And you got fencing that follows of topo. Oh my gosh, this just geeks me out totally. Isn't that awesome? Oh, I love it. Oh my goodness.

So I thank you for going on that journey with me. I really appreciate ti. One thing I should mention is that these are fence posts, but you don't have to use fence posts. You could use a wall. This could be a wall. What would you do? Let's talk about how we would do that if it was going to be a wall.

Do you remember how we had this model line-- excuse me, how we had that line in Revit on this one? All you'd have to do is do a translation up in a z to the height of your wall. And then all you have to do is take those two lines that are offset, which would be the top and bottom of your wall, create a surface through it, and then send it right back into Revit. And then you'd have a nice pretty wall that follows your topo. So that's possibly. Yes.

AUDIENCE: How about like a taper for a zone board?

MARCELLO SGAMBELLURI: How about a taper for a zoning envelope? Holy smokes, I'm talking to a civil engineer or--

AUDIENCE: Architect.

MARCELLO SGAMBELLURI: Architect, OK. Very good question. This was just to illustrate process. We had a situation where we just took a constant offset of a property line and then offset it in. I did the entire perimeter.

If you, for example, wanted to have just chunks out or it wasn't a uniform offset, you could certainly do that. You would just build in extra nodes in there to compensate for those extra offsets. But if you have any special situations, you can email me. And we'll totally geek out about it. But it's absolutely possible. Any other questions about this, before we move onto our final one. Yes, sir.

AUDIENCE: Thank you. I'm trying to [INAUDIBLE] the concept [INAUDIBLE] you have a railing element to that line and you build the railing in Revit. And it seems like having the posts and everything, [INAUDIBLE].

MARCELLO SGAMBELLURI: OK, so you mean-- So the question was, couldn't you just build railing here instead of using this post that I did? You certainly could, yes, you certainly could. I only used this simple post, because it was just a simple example. But yes, you certainly could add any type of family that you want, any level of family.

AUDIENCE: [INAUDIBLE].

MARCELLO SGAMBELLURI: Oh, the other question was, if your railing system is already defined, yes, you certainly could do that, yes. Yes. You would still need to go through the motions to get it to project onto the topo surface. But yes, you certainly could use a railing system. Any other questions? Yes.

AUDIENCE: Would the Pull Onto Surface [INAUDIBLE]?

MARCELLO SGAMBELLURI: Would the what? I'm sorry.

AUDIENCE: Pull Onto Surface.

MARCELLO SGAMBELLURI: Pull Onto Surface. Am I tell talk-- Are you a Grasshopper user?

AUDIENCE: [INAUDIBLE].

MARCELLO SGAMBELLURI: OK. The question was, would a Pull Onto Surface node work? The answer is yes. There are other ways to extract to-- there are other ways to project a two dimensional curve onto a surface.

This one was a bit more efficient to do, because the poly surface we're working with are actually little triangulated pieces. And it's break them up into little curves. So this, just in my opinion, was the best way to do, although you could go ahead and go about it that way. So there are other ways to do it. Let's talk later, and we can totally geek out about it. Yes.

AUDIENCE: How is the rotation [INAUDIBLE]? If it's a cylinder, probably it's fine. [INAUDIBLE].

MARCELLO SGAMBELLURI: OK, good question. I should have-- Good question, and then we'll move on. The question was, what about the-- You know, that's really hard to see. I apologize for that. That's not my fault. These lights are really bright.

The question was, what about the rotation. Very good question. There is a Family Instance Rotation node inside of Revit. And so you could use that if you wanted these to all align, I don't know, with the Sun angle, or whatever you want.

There's ways to do that. And me, I just put it by default of, I suppose, zero, but you could certainly change it with that node. And that node is very new. It only came in, I think, this year, Family Instance Rotation. It's awesome.

Oh, we have time. Oh, this is awesome. I'm going to get through all five. I'm so excited. Everybody ready for the last part, act three, end of act three?

How many people-- I don't know where the start with this. I'm so excited. How many people have to work with rooms? Oh, yes. How many people love the fact that they are not three dimensional elements?

So we're going to go through two big steps here. One, we're going to take those 2D rooms, we're going to turn them into three dimensional objects, so we can visualize them. Then we're going to take those 3D rooms, and we're going to put them back in the Revit, and you can do whatever you want with them. But in this case, we can create stacking diagrams.

Does anyone have to create preliminary stacking diagrams in Revit? A few of you? How do you do right now? You build masses, right, manual. I'm going to show you how to do that.

So there's two things you can get out of this-- one, changing rooms to 3D, two, creating stacking diagrams. And if you don't do it in your office, show someone who does who needs to know it. OK, here we go. So let's Open, we'll make this quick. Cool, I'm so glad I get to do this.

I set quite a bit of this up for you already. What we have are a series of levels that represent a building. This would be like the ground floor-- and, boy, that's a horrible scale. So basically, these are separate rooms, basically layouts that you would be doing. This is parking, or interior. You have ventilation, retail, all that kind of stuff.

Also these are color-coded. This one's red in this planting and so on. So you can do this for preliminary studies or even if you're right in the middle of your project. This would be your conference space and so on and level two. You have these all stacked up-- conference space and so on.

So I'm going to show you how to actually extract this data and turn it into 3D elements in Revit. And it's very similar to what we've been doing in the past. So the first thing you have to do is you got to use start a new Dynamo definition. And then you want to be able to select all the rooms out of Revit. And in order to do that, you want to Select by Category.

So we'll go ahead and go Rooms, right here. Don't worry, we'll get in time Rooms. And then we say, All Elements of Category. We pull all the rooms.

Now, do you notice you don't see anything in 3D? Why don't you see anything in 3D? Because rooms aren't 3D, right? Yeah, what a mess. But anyway, we're better than that, right? We're going to make them 3D.

So then what we do is we pull all the rooms. Then all we need to do-- do you remember, if you want to visualize something, you want to change anything into Dynamo Geometry, you go ahead and go here to Geometry.

And then under Geometry-- oh, wait. No, I'm in the wrong spot, am I? Revit, and then Element, and then under Element, Element again. And then you can pull the Geometry. Or you can actually pull-- we know that rooms are actually made of solids.

So I'll fire this up. And this one's beautiful, because when you run this, Dynamo will actually interpret all of your rooms based on the heights you give them. And it will turn them into Dynamo solids.

How cool is that? That's awesome, right? Yeah, pretty cool, huh? So you can start to visualize your rooms inside of Revit.

You can take it a step further. Now, this is only Revit Geometry, isn't it? There's a brand new node on the block. And it's something beautiful. Let me show it to you.

It's called DirectShape. DirectShape allows you to take Dynamo Geometry that you have created very meticulously and stick it back into Revit in any category that you want. It's cool. So all you do is type in DirectShape. OK, DirectShape, it's here somewhere. DirectShape, here it is, by Geometry.

Don't worry, we'll end on time. Here we go. So you just feed in the Geometry. The Category, we'll make it a Room Category. And the Materials, in the Materials, what you can do with Materials is, there's actually-- I actually preset all the materials to have the same name as the room name. But you don't have to necessarily do that.

There's an element called-- there's a node called Material By Name. If you feed in the-- OK, actually, we got to pull the name. So we got to go here to Get Parameter. We got to feed in this one. And then we got to say, Get The Name, like this.

So name of-- whoops. So this is all Name of the Room. Here we go. And then we feed that in. We should get the room names. OK, good. And then we feed that into the Material name. I'll Run it again, make sure we get all the materials too. Don't crash on me.

Come on now. Oh, you're going to crash on me. Oh, here we go. OK, cool. That's why. Didn't have it in there.

And then all we do is feed the-- wasn't working. OK, let's just run default for now. And then if Material-- if this is unnamed, it'll send the geometry back into Revit. But you could color code this. And this is in the handout. I'm not sure why that Material node wasn't working, but I want to get this geometry back in.

So the DirectShape takes any geometry, any category, and puts it in there. So as it chews on this, let me tell you how powerful that is. What if you built a road that follows topography in Dynamo and then you wanted to take that solid and put it back into Revit? With DirectShape-- once it stops chewing-- DirectShape will actually place it in any category. So if you want to put a, road, you can actually place it into the Road category that was always this mysterious thing that no one really ever knew about.

So did it work? Oh it did, but it gave me a funny material, didn't it? I got one minute. I got 58 seconds. OK, let's see if I can do this. Let's see.

The Material, if you don't set the Materials just right, it'll put it on this funny default Dynamo one. I'm not sure why. Here it is. But the handout will show you how to actually change the colors and all that. But we won't go through that now. So we can do that.

See, and it actually sends it right back in. And then you've got a 3D stacking diagram. Well, this one is not color-coded, but do you see? You actually have these inside of Revit now, and you have 3D rooms inside of Revit. How cool is that? Pretty awesome, huh?

[APPLAUSE]

Yeah, OK, cool. So we're coming in for a landing. I appreciate everyone for showing up. I'll be around all week. And if you want to geek out about this, I'd love to talk to you about it. I love that you were here.

You in the audience is what makes me do this. Love helping you. I love talking about this. So I want everyone to be a Dynamo hero out there. You go out there, and I want you to save some money, and I want you to help people out.

And if you have any other questions, just contact me. And thank you, everyone. Go out there and do awesome things. Thank you.

[APPLAUSE]