



REVIT PURE PRESENTS  
**PAMPHLETS**



**ISSUE #22 / FALL 2021**

# DYNAMO



[revitpure.com](http://revitpure.com)

© Copyright 2021 - BIM Pure productions  
by: Nicolas Catellier, Architect

# PAMPHLETS COLLECTION



ISSUE #1  
**WORKSETS**



ISSUE #2  
**SCHEDULES**



ISSUE #3  
**PHASES**



ISSUE #4  
**LINK CAD**



ISSUE #5  
**FILTERS**



ISSUE #6  
**VIRTUAL REALITY**



ISSUE #7  
**3D VIEWS**



ISSUE #8  
**PLAN NOTES**



ISSUE #9  
**COORDINATES**



ISSUE #10  
**SCOPE BOXES**



ISSUE #11  
**FINISHES**



ISSUE #12  
**LINE WEIGHTS**

# PAMPHLETS COLLECTION



ISSUE #13  
**DOORS**



ISSUE #14  
**SHARED SITES**



ISSUE #15  
**pyREVIT**



ISSUE #16  
**FREE ADD-INS**



ISSUE #17  
**SLANTED WALLS**



ISSUE #18  
**RAILINGS  
PART 1**



ISSUE #19  
**RAILINGS  
PART 2**



ISSUE #20  
**PHASES  
PART 2**



ISSUE #21  
**LANDSCAPE**



# PAMPHLETS

---

ISSUE #22 / FALL 2021 / DYNAMO

## WHAT IS THIS "PAMPHLET"?

Revit Pure Pamphlets are published 4 times a year by email. Each edition covers a particular Revit theme. We like to pick themes that are complex and confusing. Our job is to make these topics simple for you.

## WHY DYNAMO?

If you haven't started to use Dynamo in your Revit practice, I can confirm you are wasting hours of your precious time.

Is there a boring, repetitive and tedious task you find yourself doing over and over again? Dynamo can almost certainly automate it for you.

This is the perfect starting point if you don't have any experience with Dynamo. This is a kindle to light the automation fire in your mind. Once you get going with Dynamo, you'll ask yourself why you didn't start sooner.

Thanks and good luck :)

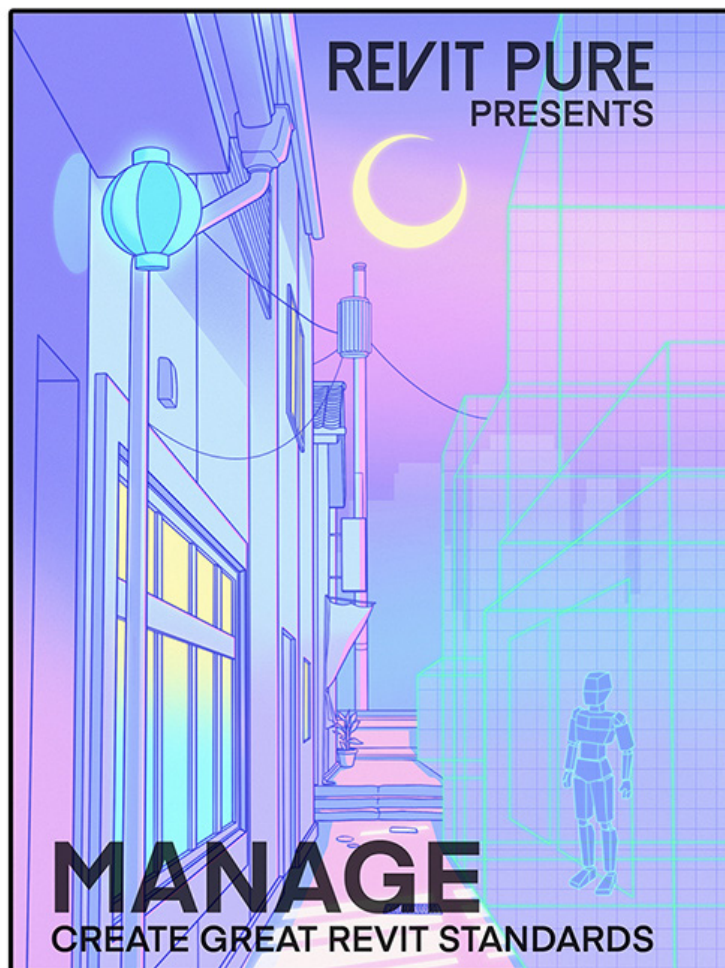


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## MANAGE LEARNING PACKAGE

This pamphlet is an excerpt from our recent MANAGE learning package for Revit. Do you want to get the complete Dynamo chapter, plus 16 other chapters to master Revit? [Click on this link](#) to get the entire course. Pamphlet readers get 15% off.





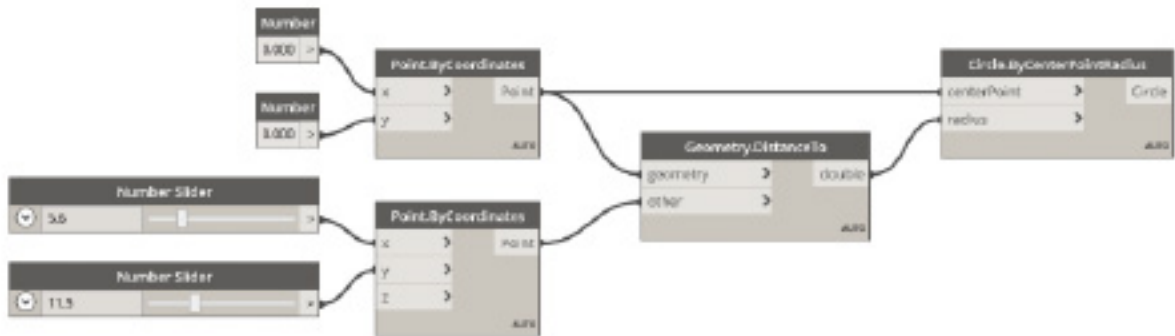
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## WHAT IS DYNAMO?

Dynamo is a visual programming platform for Revit. Typically, programming is done with text. Visual programming replaces code with nodes and wires instead. In the end, you can achieve the same result by building a custom add-in than a Dynamo script. In the image below, the **Visual** and **Textual** programs both accomplish the same thing.

### Visual Program:



### Textual Program:

```
myPoint = Point.ByCoordinates(0.0,0.0,0.0);  
x = 5.6;  
y = 11.5;  
attractorPoint = Point.ByCoordinates(x,y,0.0);  
dist = myPoint.DistanceTo(attractorPoint);  
myCircle = Circle.ByCenterPointRadius(myPoint,dist);
```

In this chapter, we will focus on modifying parameters and information. Dynamo can be used to generate geometry, but these features are usually more complex. We'll focus on the simple stuff first.

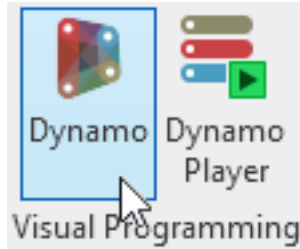


# PAMPHLETS

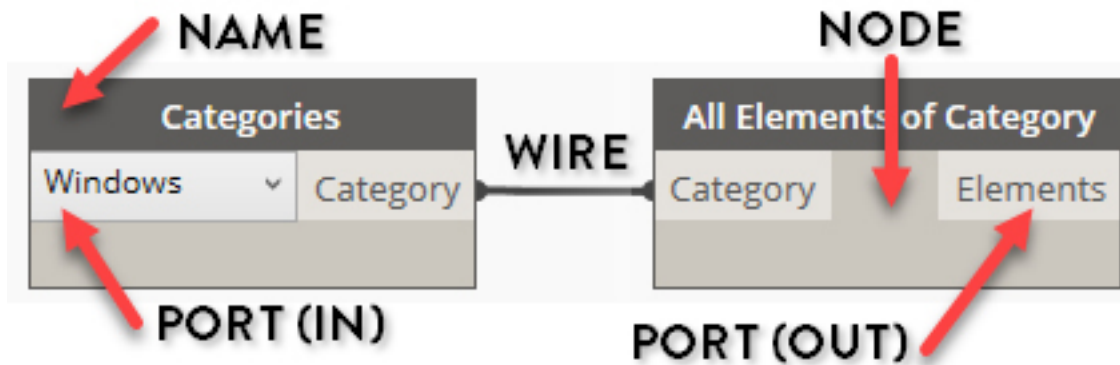
ISSUE #22 / FALL 2021 / DYNAMO

## DYNAMO TERMINOLOGY

Open Dynamo. You will find the tool on the ribbon, in the **Manage** tab.



Here is the basic terminology of Dynamo. **Nodes** are the little boxes you keep seeing all around. We'll use that word a lot.



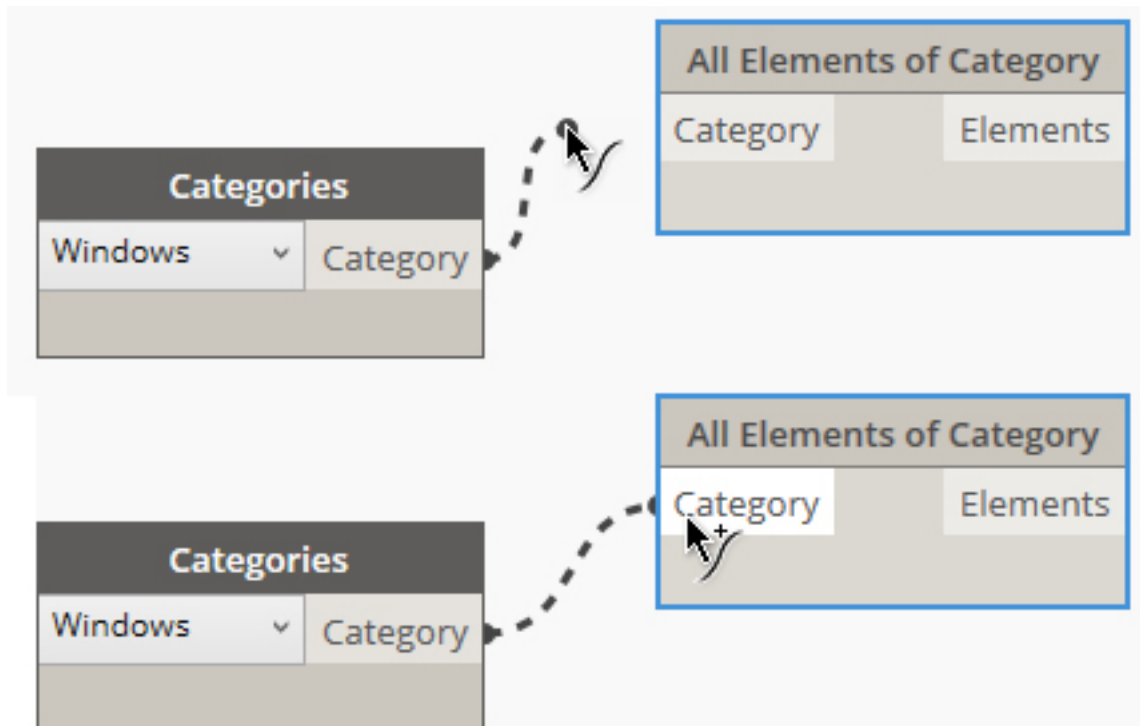


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## CONNECTING NODES

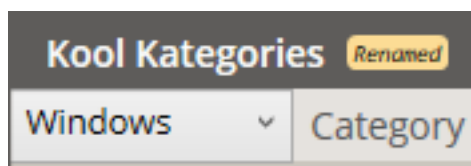
The core idea of Dynamo is to connect nodes using wires. That's what we do in the image below.



## CONNECT NODES WITH WIRES

## RENAMING NODES

Double-click on a node to rename it. Renamed nodes will have a yellow "Renamed" tag alongside the name.





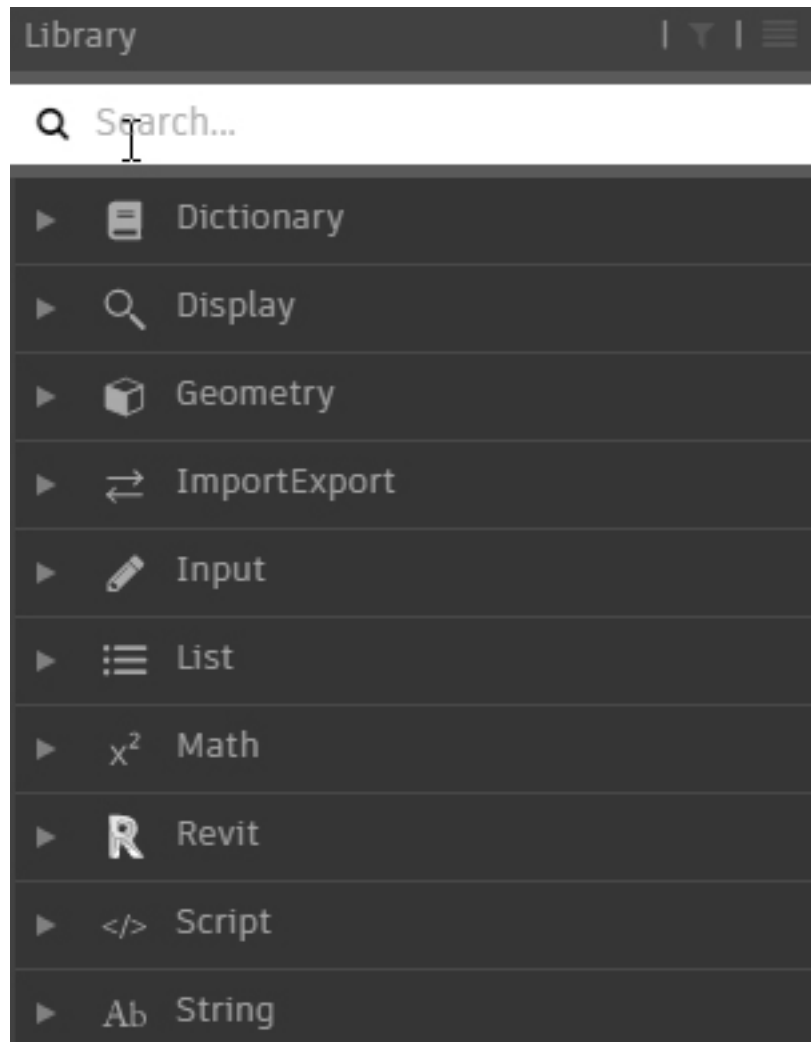


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## NODES LIBRARY

On the left of the interface, you'll see the node library. They are divided into multiple categories. The nodes that interact with Revit are part of the **Revit** category. The search bar is essential to find your way around the huge number of nodes.



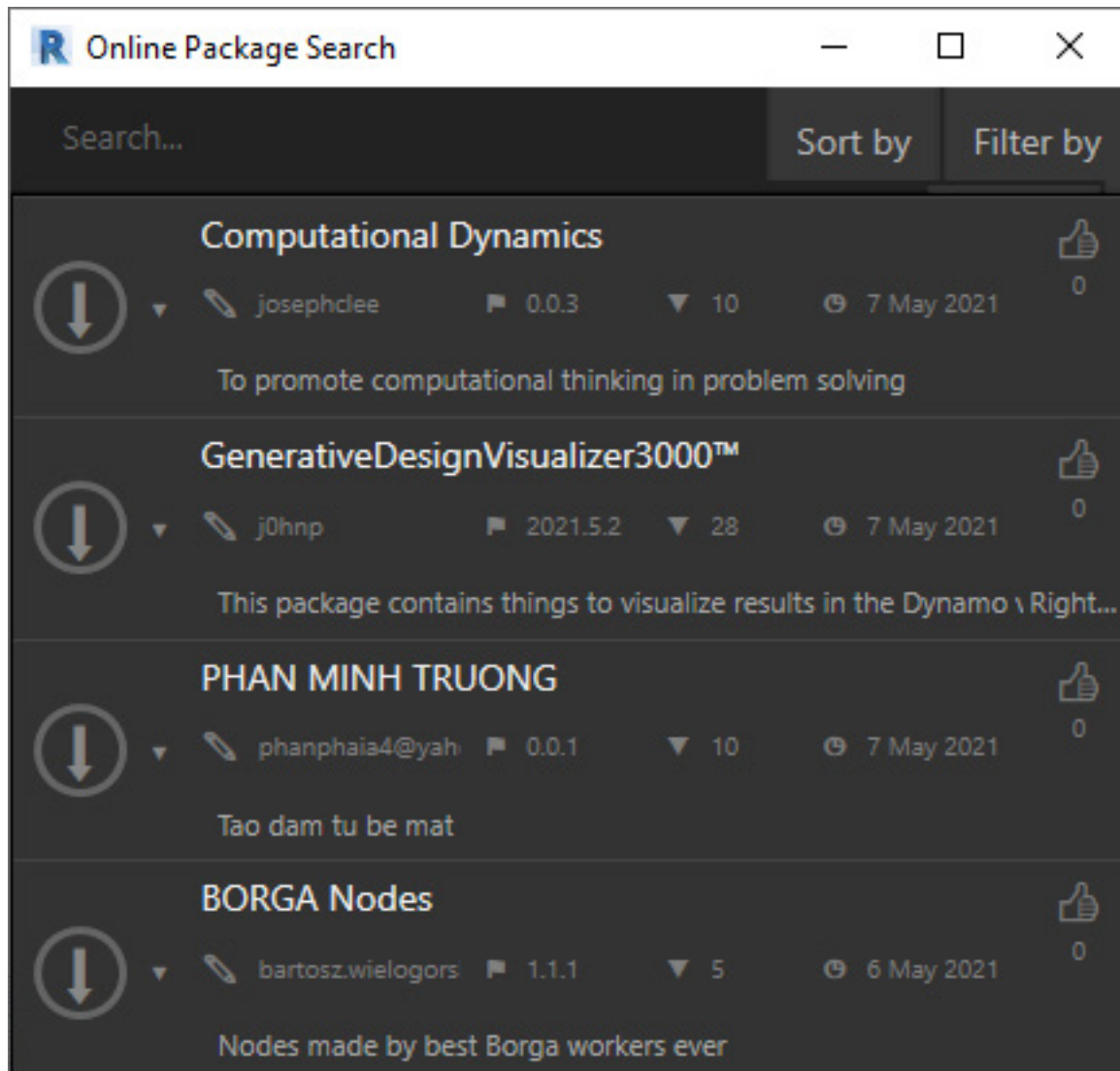


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## CUSTOM COMMUNITY PACKAGES

A strength of Dynamo is the active community which keeps providing additional nodes for free. Go to the **Package** menu and select **Search for a package**. Some packages are essential for a great Dynamo experience. More on that later.



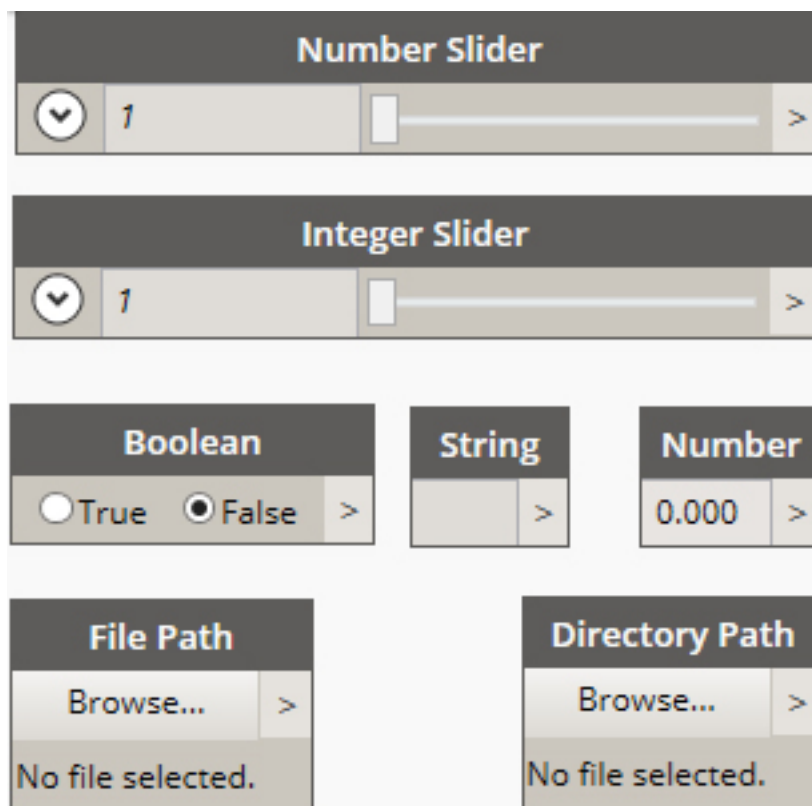


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## INPUT NODES

Input nodes are essential to get your script started. They are used to define specific numbers, text, file path, yes/no and much more. Here are some input nodes you'll encounter:



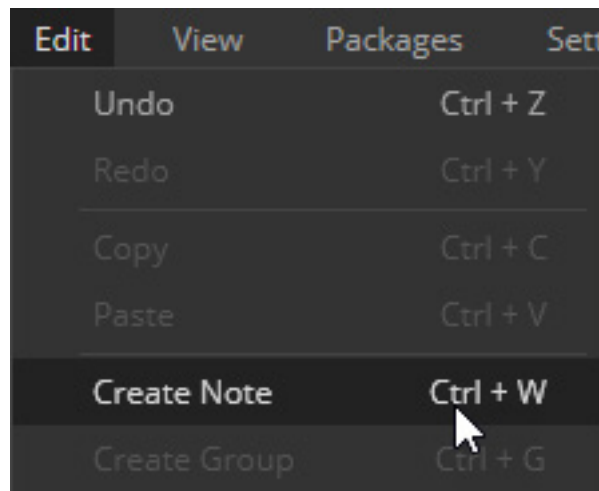


# PAMPHLETS

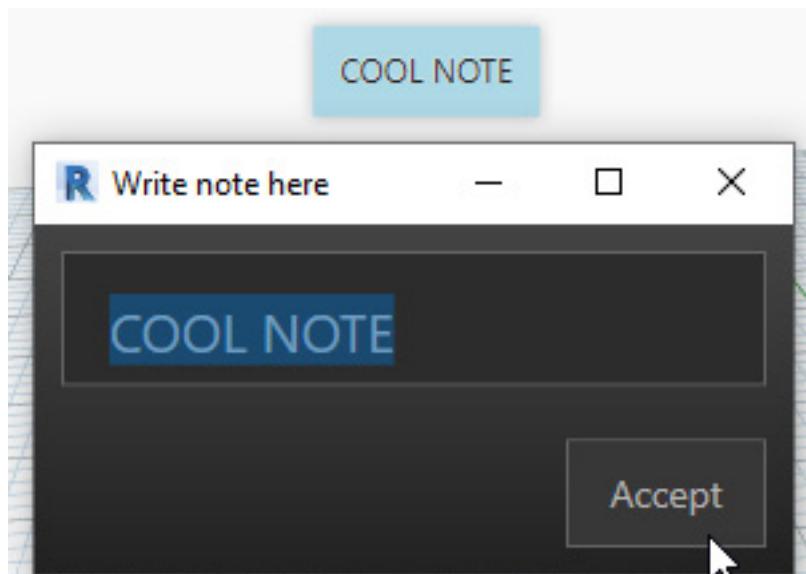
ISSUE #22 / FALL 2021 / DYNAMO

## CREATING A NOTE

Some scripts can become insane spaghetti festival. Adding notes to the script helps other user understand what's going on. Go to **Edit** and click on **Create Note**.



Then, type in whatever you want and click Accept.



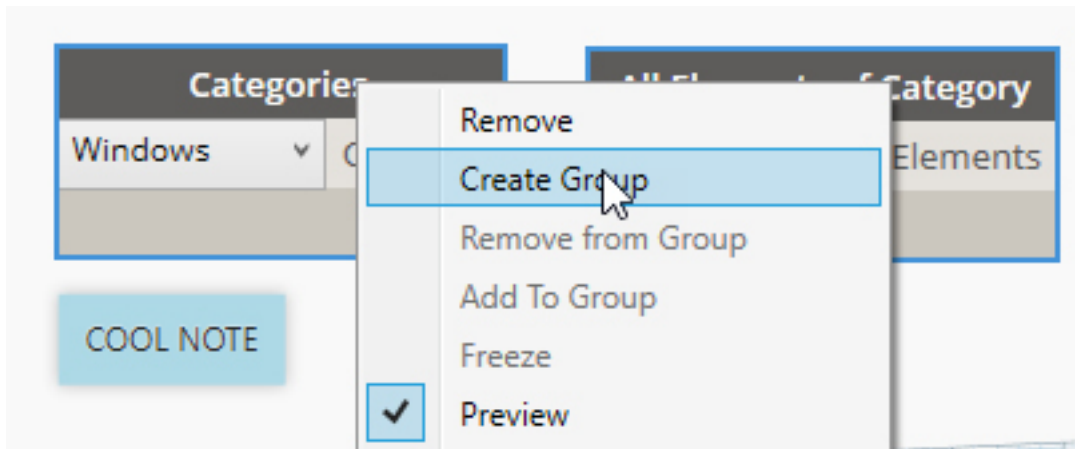


# PAMPHLETS

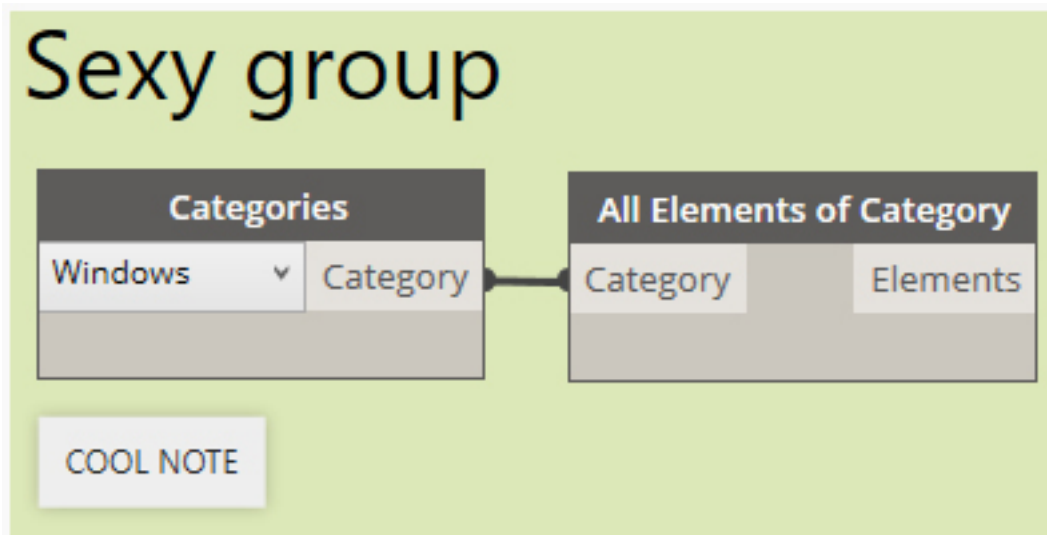
ISSUE #22 / FALL 2021 / DYNAMO

## CREATING A GROUP

Notes are okay, but groups are better. Select multiple nodes and notes. Then, right-click and select **Create Group**.



Check out this awesome group:

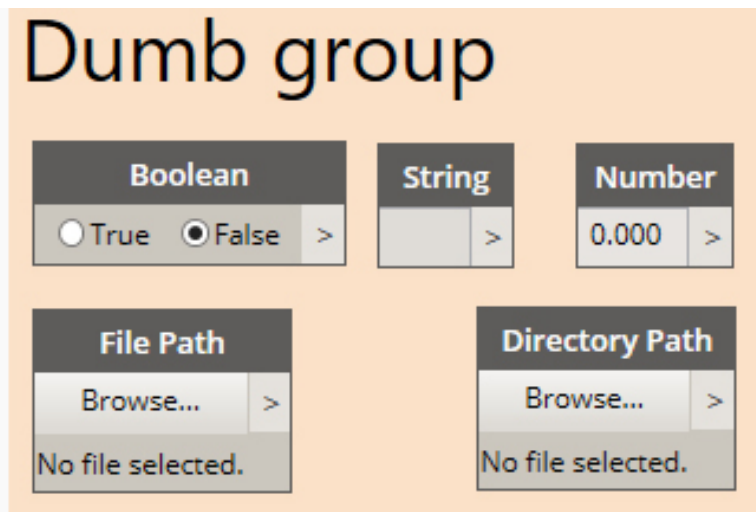
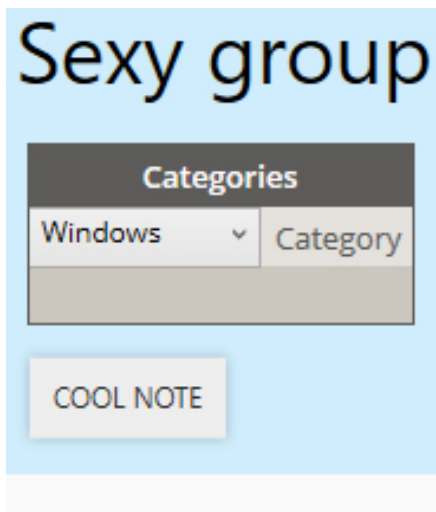
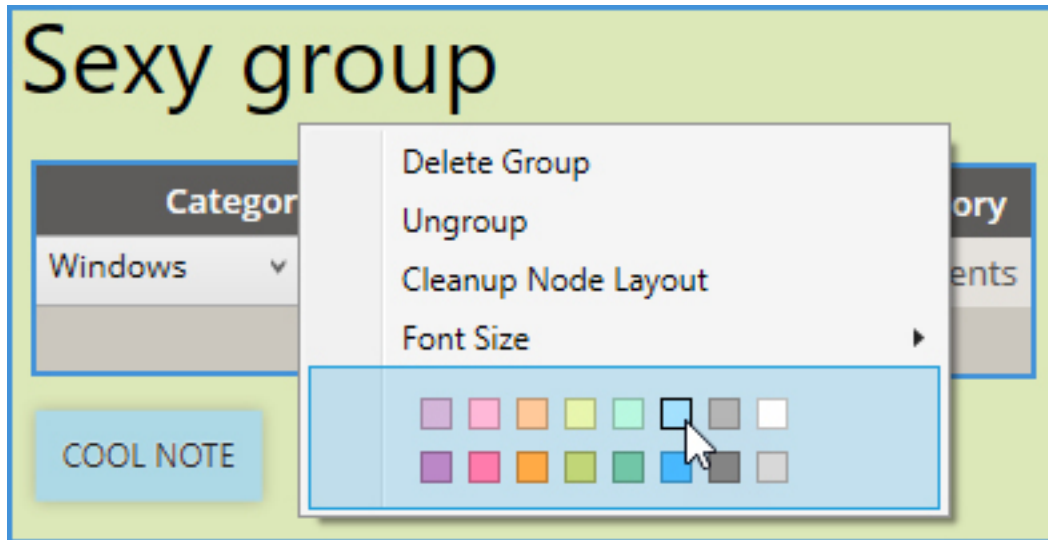




# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

You can show a group color by right-clicking it. Multiple colors will be available. When creating big scripts, use a strong color code to keep things easy to manage.



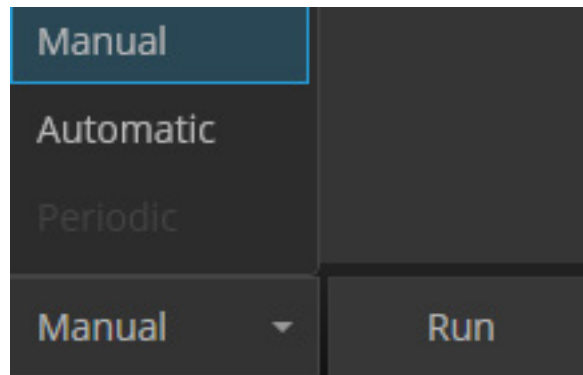


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## MANUAL MODE VS AUTOMATIC MODE

The Automatic/Manual option is located at the bottom right of the interface.



Automatic mode will run and rerun the script each time you modify it. This is the default mode in Dynamo, but probably not convenient for you. Automatic mode often execute scripts before users are ready to do so. You can potentially cause problems to your model if you aren't careful.

Instead of **Automatic**, switch to **Manual** instead. When you want to activate the script, click on **Run**. Regardless of which mode you are using, create a test model before using your script on an important Revit model.

## DYNAMO PLAYER

Some users are terrified of opening Dynamo. That's ok, they can use the Dynamo Player to run the script and they'll never face the spaghetti.



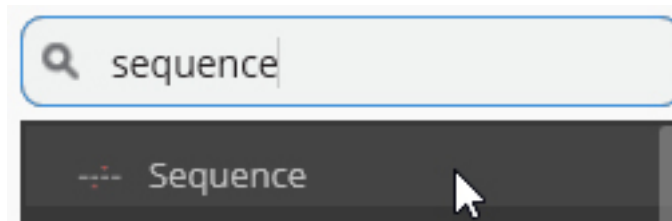
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## CREATING A SEQUENCE

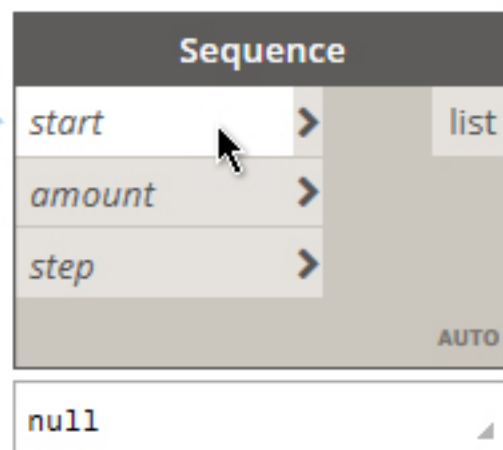
That's enough with the interface, let's create something.

Add a sequence node. Use the right-click menu and type in **Sequence**.



Hover your cursor over one of the input ports. You will see a pop-up that explains what that input does and what input type the node expects—hovering your cursor above the entire node will explain how to use the node.

Number or letter to start the sequence at  
Default value: 0







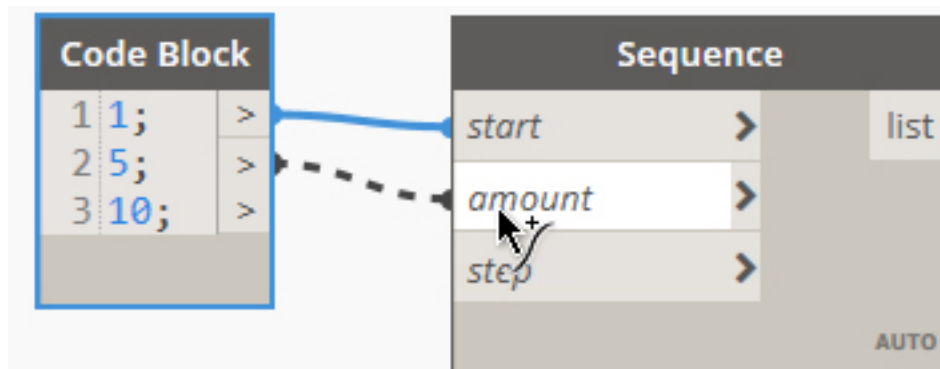
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## ADDING INPUT NODES

Let's explore a few ways to create input. The universal Dynamo node is called **Code Block** and can be used for multiple things. You can indeed type in code, but also simply type in values. To create a code block, double-click anywhere on the screen.

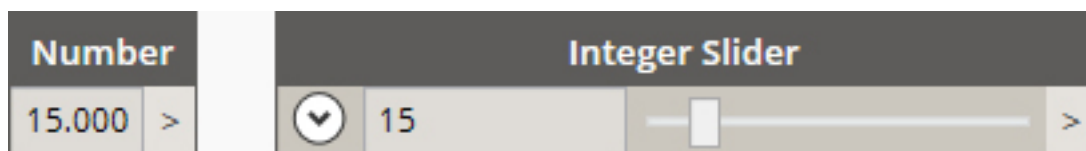
In the example below, we type numbers. Press **Enter** to create multiple lines on the same Code Block. Plug the wires.



Code Block has some limitations. One of them is that the values entered here can't be customized in the Dynamo Player.

## NUMBER VS INTEGER

The Code Block alternative is to use a **Number** or **Integer** node. If you remember your math class, an integer is a number that is not a fraction.



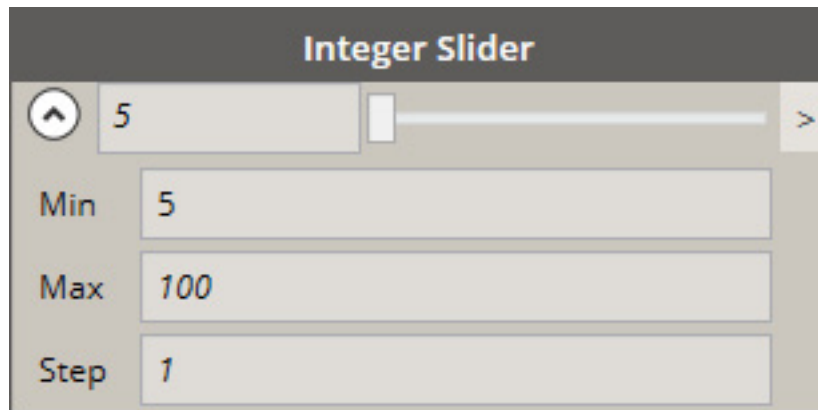


# PAMPHLETS

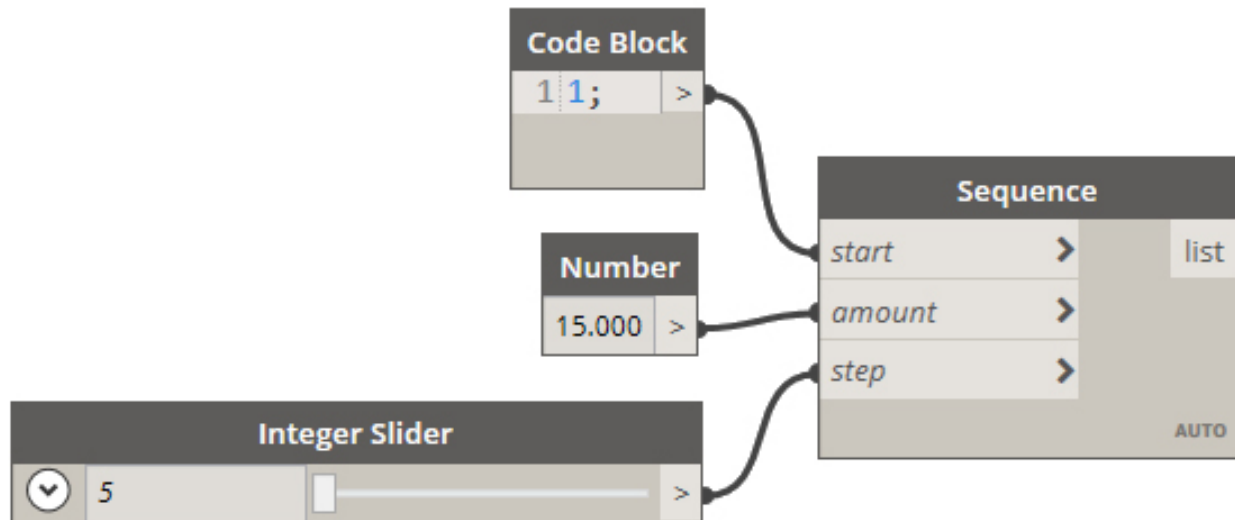
ISSUE #22 / FALL 2021 / DYNAMO

## USING A SLIDER

When creating number or integer input, you can use **sliders**. Click on the arrow symbol to adjust min, max and step.



Let's plug that slider into the sequence!





# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## EXECUTING THE SEQUENCE

Execute the script. The sequence is generated. Hover your cursor on the node and you'll see a **List** pop-up. Go to the list and click the **Pin** icon so you can see all items on the list permanently.

The image shows two screenshots of a Dynamo script interface. The left screenshot shows a 'Sequence' node with parameters 'start', 'amount', and 'step', and a 'List' output node showing '{15}'. The right screenshot shows the 'List' node expanded to show a list of 15 items, with a 'Pin' icon in the top right corner of the list area.

Index	Value
0	1
1	6
2	11
3	16
4	21
5	26
6	31
7	36
8	41
9	46
10	51
11	56
12	61
13	66
14	71



# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

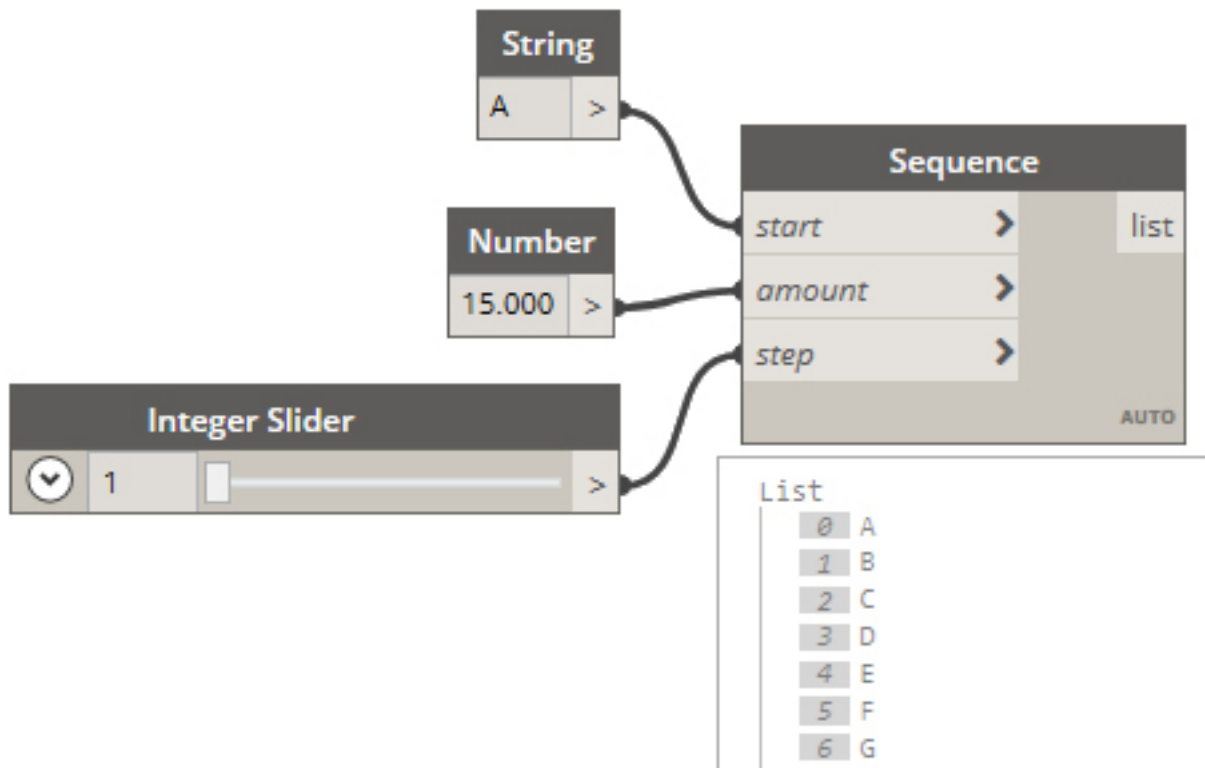
## USING A "STRING"

Time for some programming jargon: string means text.

Create a string node.

Type in a letter. As you can see below, the script now creates an alphabetical list instead of a number sequence.

This specific type of entry only accepts a single letter, but other nodes will allow you to use as many words and letters as you want.





# PAMPHLETS

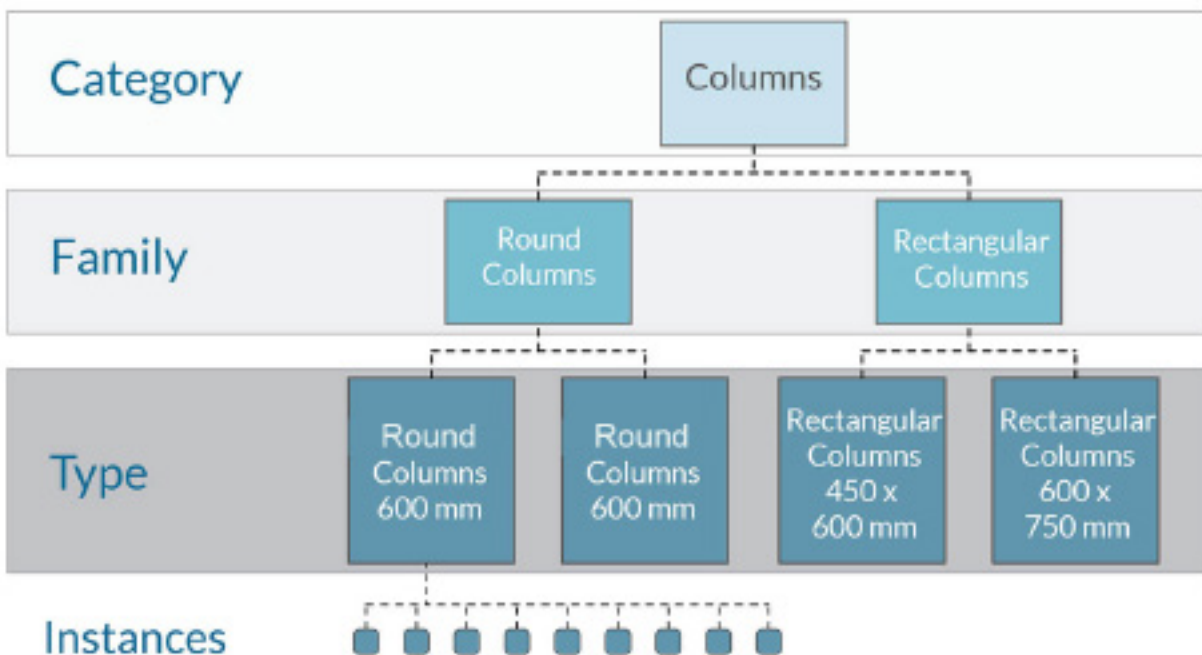
ISSUE #22 / FALL 2021 / DYNAMO

## 12 STEPS TO CREATE YOUR FIRST DYNAMO SCRIPT

Ok, we've created a sequence, but that doesn't amount to much. It's time to get serious and create a script that will allow you to modify the parameters of specific elements.

### 1- UNDERSTAND DIFFERENCE BETWEEN CATEGORY, FAMILY, TYPE AND INSTANCES

You are ready to interact with a Revit model. First, take a look at this chart and make sure you understand it. Else, you'll get confused with nodes.





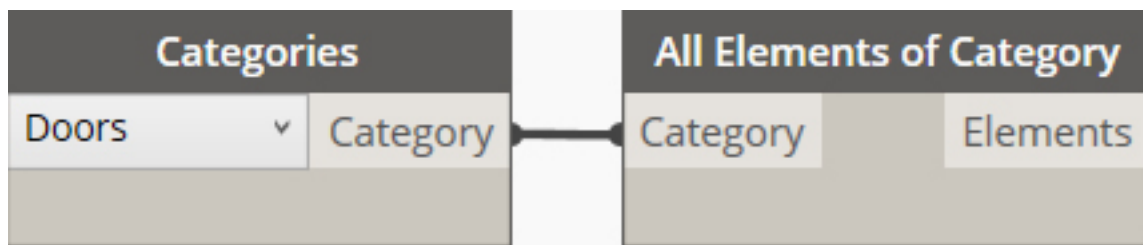
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

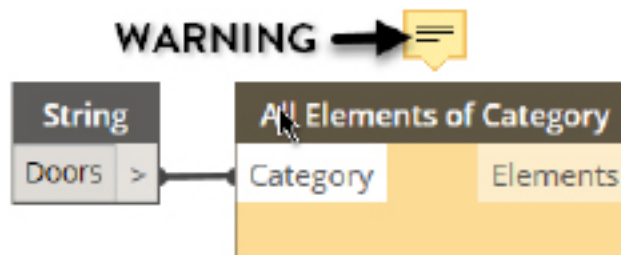
## 2- DEFINE ALL ELEMENTS OF A CATEGORY

Let's say you want to pick all doors in your project. You'll need to bring two nodes: **Categories** and **All Elements of Category**. Use the search bar to find the nodes.

In the Categories node, use the dropdown menu to pick a category. In this example, we use **Doors**.



A quick note here: if you try to type Doors in a String or Code Block node, you will get a warning. Revit expects the **Category** input type. A string isn't a category. That's why you need to use the Categories node.





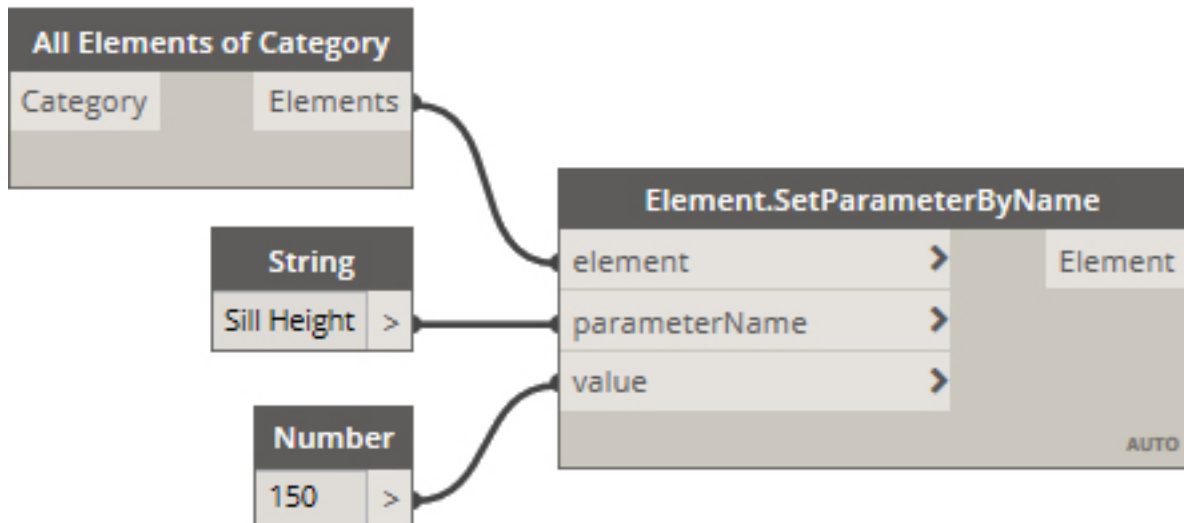
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

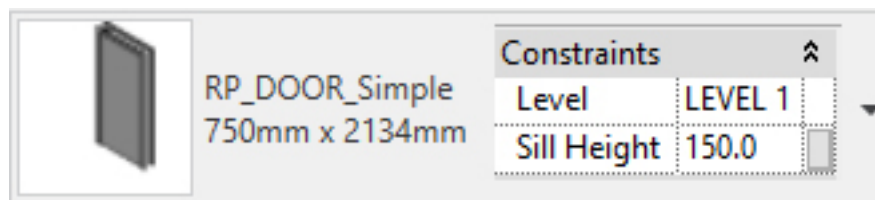
## 3- ADD "Element.SetParameterByName" NODE

Search for the **Element.SetParameterByName** node. Set the doors to the element input port. Use a string value for the **parameterName** and a number for the value.

For the sake of this exercise, we use the **Sill Height** parameter to modify doors.



When your script is ready, click on Run if you are in Manual mode. If you are using Automatic mode, the script will be executed automatically. Check out your doors. The Sill Height should be adjusted to 150mm:





# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

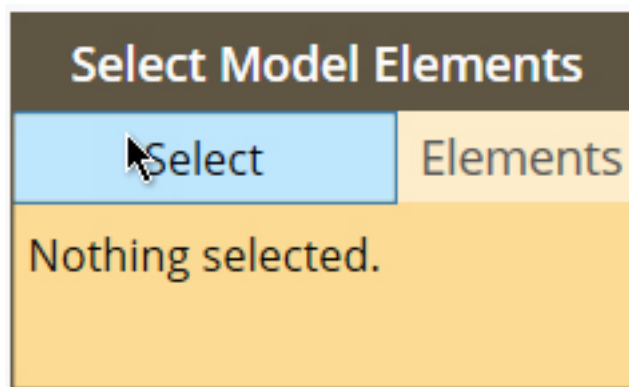
## 4- SWITCH TO THE “SELECT MODEL ELEMENTS” NODE

Let’s just pause for a moment and celebrate the fact that you’ve created your first useful Dynamo script. Isn’t that amazing? You can easily flip a few nodes to potentially affect all parameters of all categories in your projects.

We’ve used All Elements of Category in this case, but what if you want a specific elements selection instead?

Bring in the **Select Model Elements** node. Be careful: another node called **Select Model Element** (singular) only allows you to select a single element.

When you click on **Select**, go back to Revit and do a window selection.



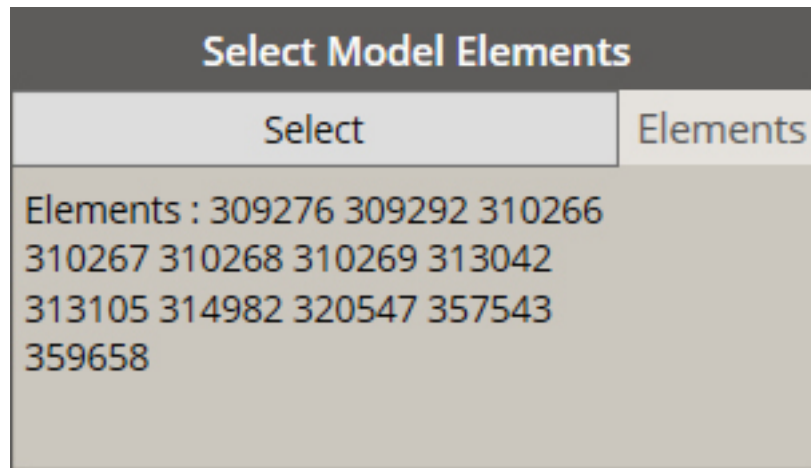




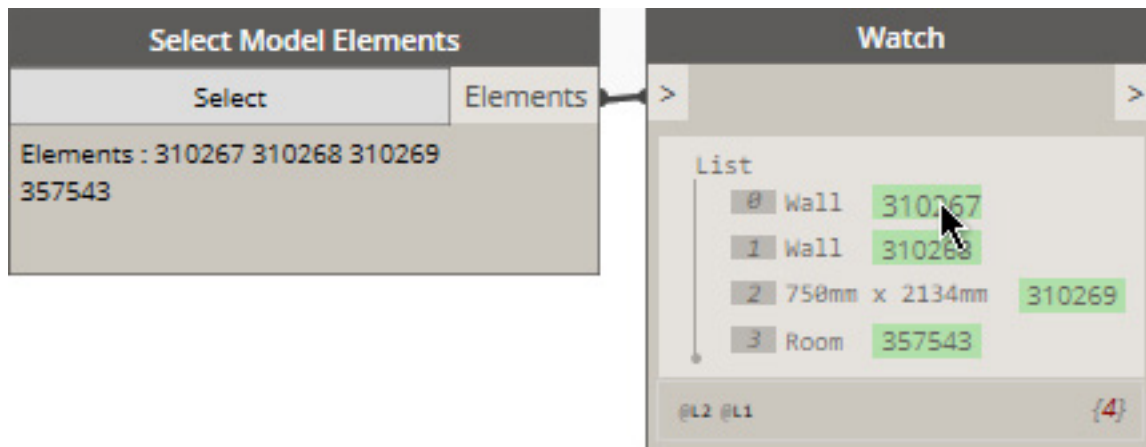
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## 5- UNDERSTAND ElementID AND WATCH NODE



When elements are selected, you will see a bunch of numbers populate the node. These represent the **ElementID** value, which is a unique number for each element. Add a **Watch** node: it will indicate the category or type of each unique ElementId. Click on the green ElementId to show the element in your Revit model.



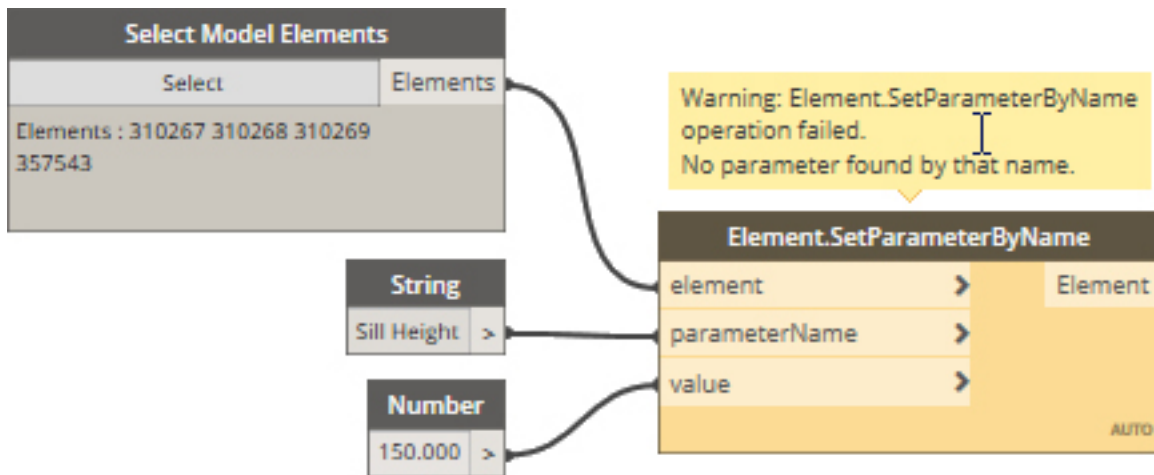


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## 6- CREATE A SELECTION FILTER

Try to execute the script with your selection: you'll most likely get a warning. That's because you selected elements from multiple categories, not just doors. Other categories don't have a **Sill Height** parameter, resulting in a Dynamo warning. The script is still working, but it is inefficient. This action isn't good practice: you need to clean your script.



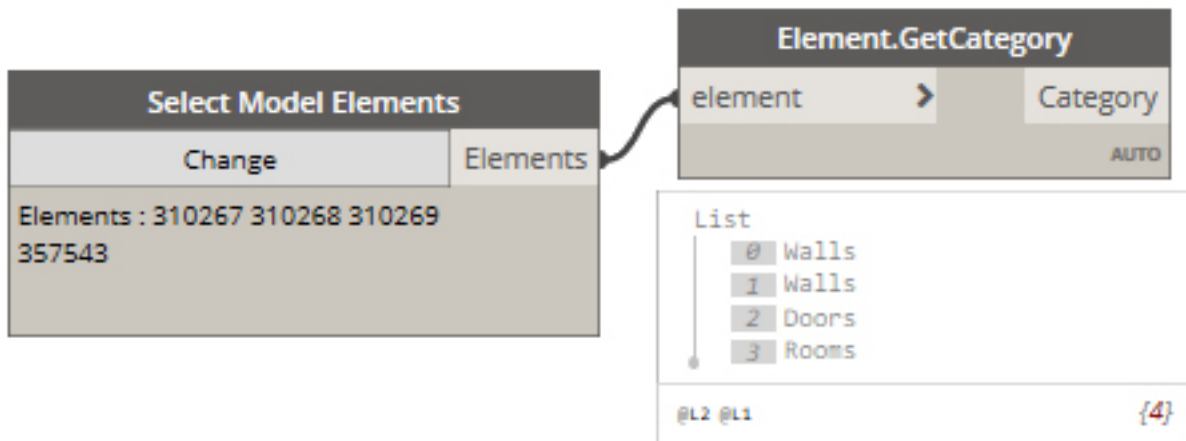
**YELLOW NODE = WARNING**



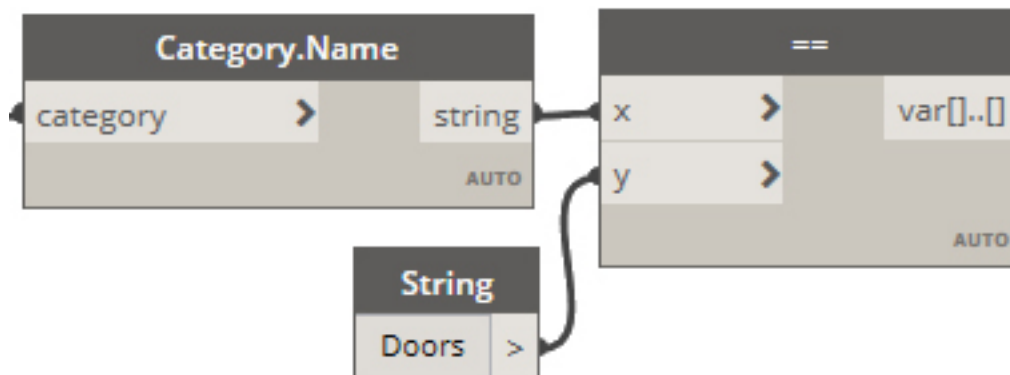
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

Let's bring a node called **Element.GetCategory** and plug it in. The node creates a list with the category for each element.



Let's bring in another friend to the party. This one is called **Category.Name**. It will convert the categories to strings. Then, bring in the node called **==**. This node will compare two values and give you a list that contains either the **True** or **False** values.



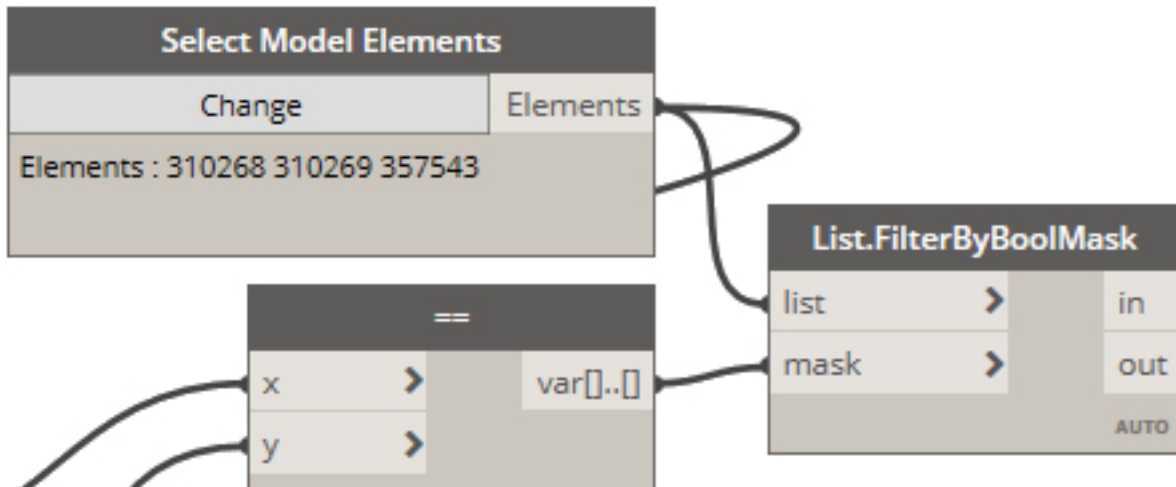
In the **x** input, have a list with the category of each selected element. In the **y** input, we plug in a **Doors** string input. The **==** node answers the question: is  $x = y$ ? If yes, it returns **True**. If no, it returns **False**.



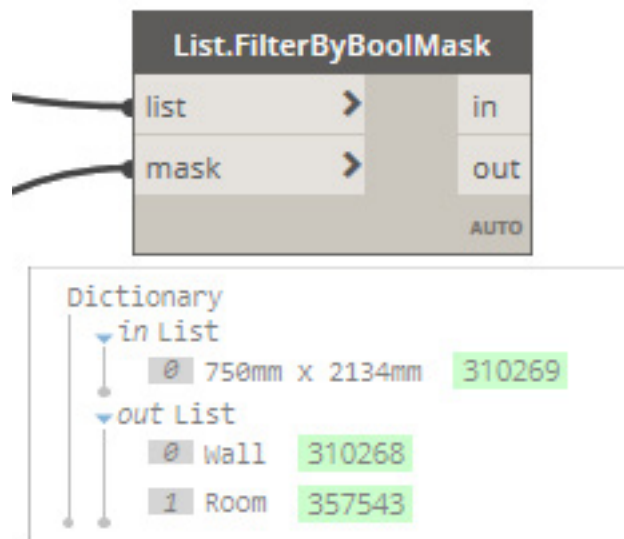
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

To properly use the BoolMask node, you have to plug-in the original selected elements to the list input. Plug the result of the == node in the mask input. In the example below, you can see the **Select Model Elements** output is connected to two different nodes.



When the script is executed, the **in** list contains all doors, while the **out** list contains elements from all other categories.

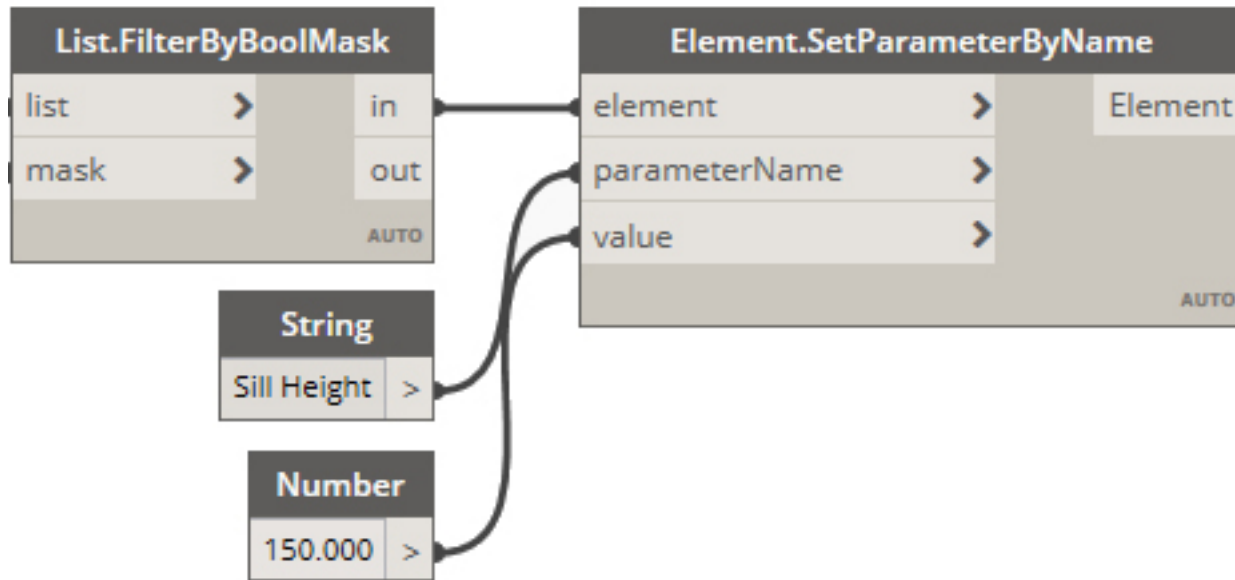




# PAMPHLETS

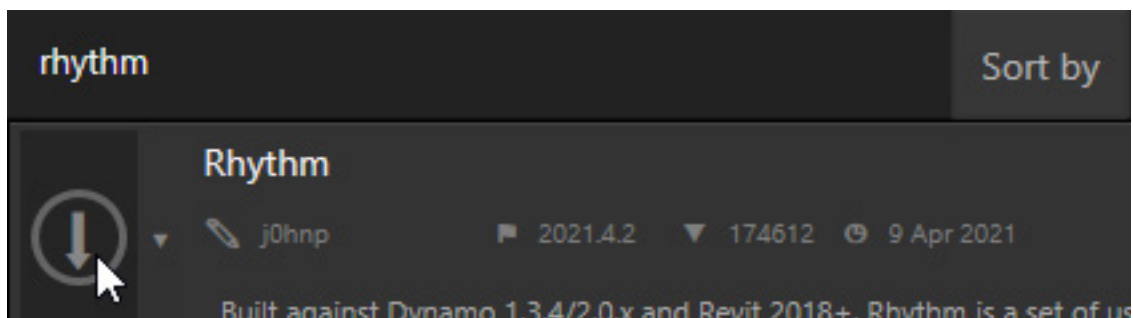
ISSUE #22 / FALL 2021 / DYNAMO

Now, plug in the in list to the **Set.ParameterByName** node you've added earlier. No more yellow warning!



## 7- ALTERNATIVE: USE NODE FROM RHYTHM PACKAGE

Let's be honest: that was a lot of nodes just to filter by category. Let's introduce John Pierson's amazing Rhythm package. It contains the perfect node for you. First, search for the Rhythm package like you've learned in the previous section. Click on the arrow button to download.





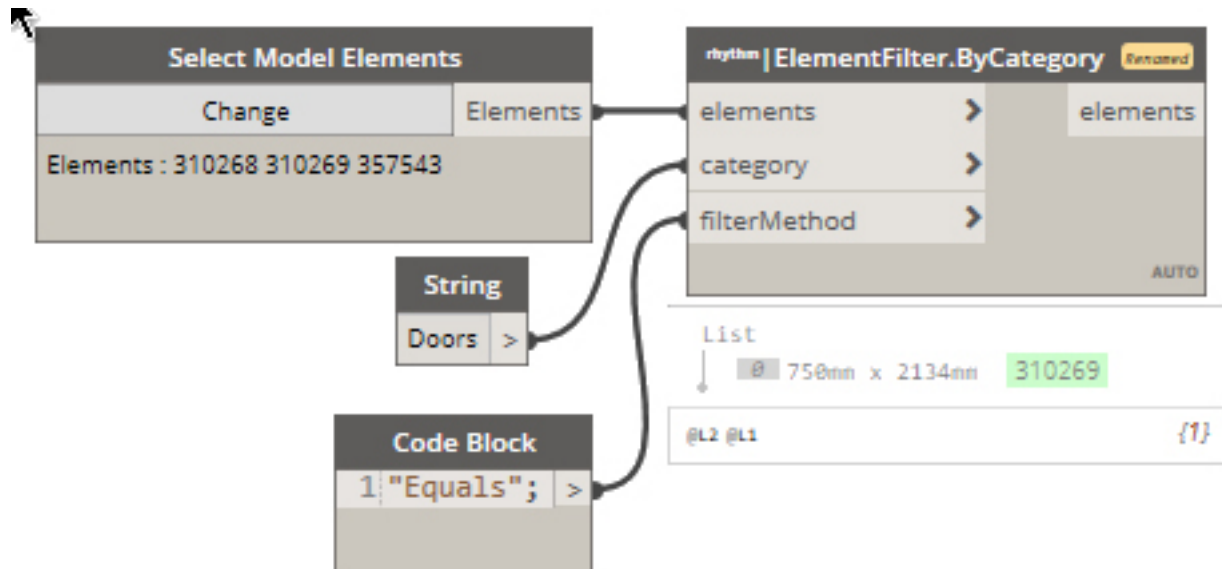
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

Now, add the node called **ElementFilter.ByCategory**. If you are confused by the node, hover your cursor on the inputs to get full explanations.

*Note: the search tool in Dynamo can be capricious. If you can't find the node, try typing in "ByCategory".*

This custom node replaces the 4 default nodes we had to use in step #6. Big thanks to John Pierson, the oracle of Dynamo.



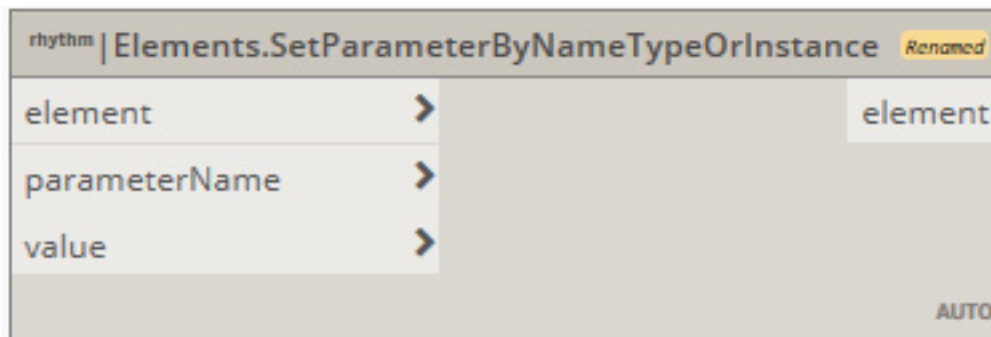


# PAMPHLETS

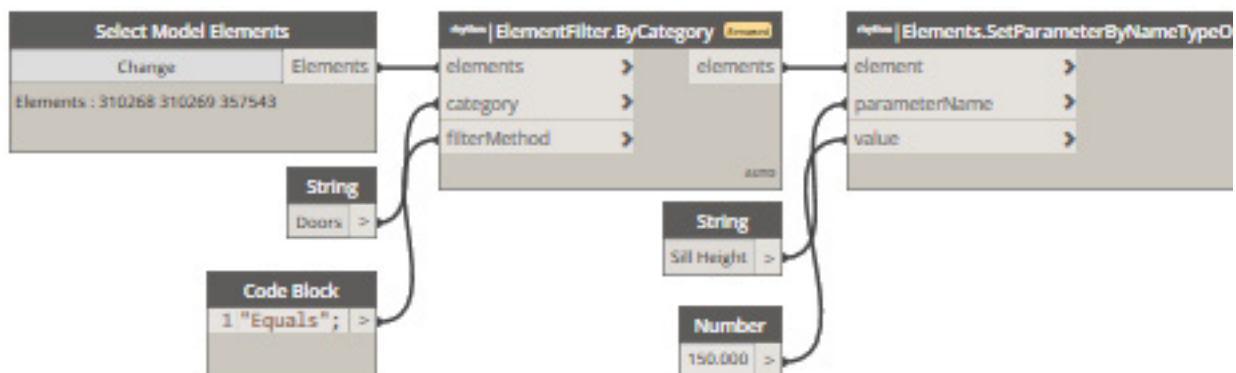
ISSUE #22 / FALL 2021 / DYNAMO

## 8- REPLACE "SETPARAMETERBYNAME" NODE

The Element.SetParameterByName is limited to instance properties. The Rhythm package contains another helpful node called **Element.SetParameterByNameTypeOrInstance** that also works with type parameters. Replace the old node with this custom node.



This is what the final script looks like.





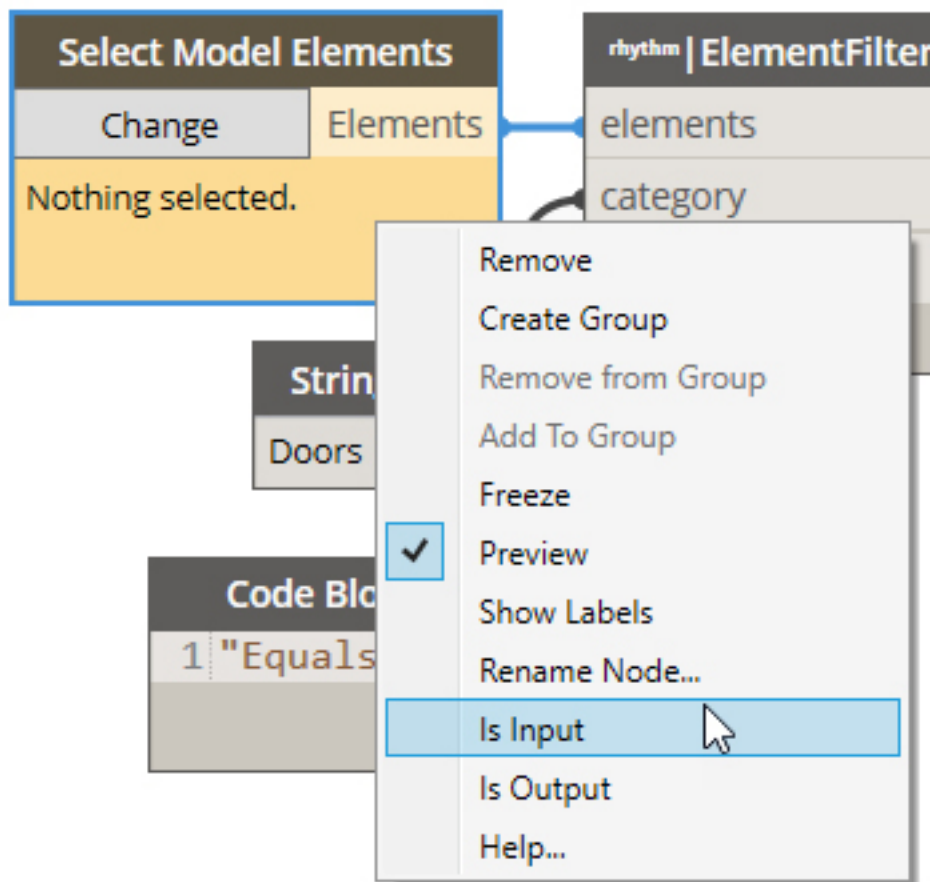
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## 9- PREPARE THE SCRIPT FOR THE DYNAMO PLAYER

The dynamo player is great if you want people in your team to use the script without interacting with the spaghetti interface.

There are a couple of easy steps to take if you want to bring your script to the Dynamo Player. Right-click on nodes and activate "Is Input". Repeat the process for all input nodes you want to access in the Dynamo Player.



**ACTIVATE "IS INPUT"**

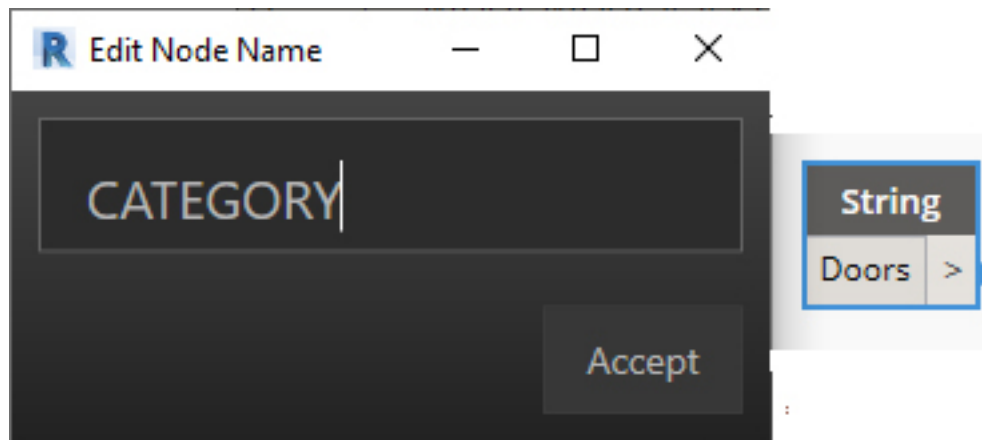




# PAMPHLETS

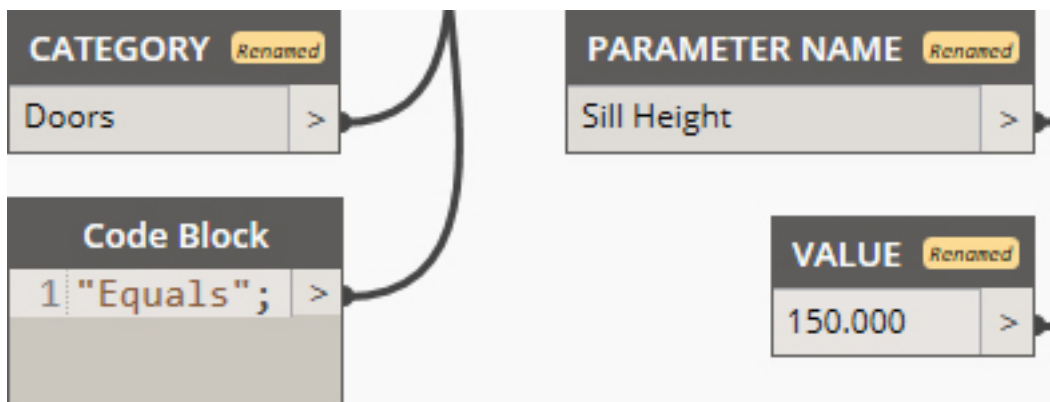
ISSUE #22 / FALL 2021 / DYNAMO

Once the nodes are set to inputs, you have to **Rename** them. Double click on a node and set a name that will be easy to understand for all users. In the example below, we rename the String node to CATEGORY.



## RENAME INPUT NODES

In this script, we rename 3 nodes for Category, Parameter Name and Value.



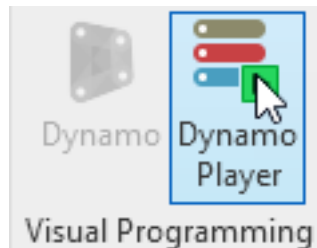


# PAMPHLETS

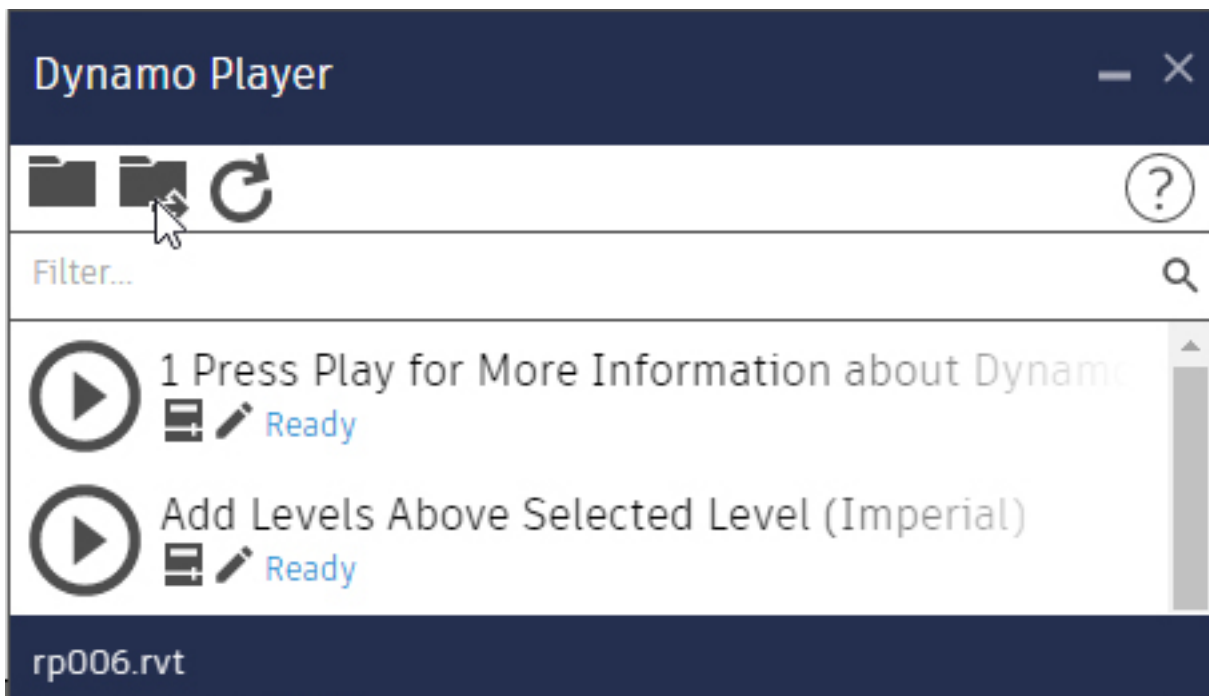
ISSUE #22 / FALL 2021 / DYNAMO

## 10- BRING THE SCRIPT TO THE DYNAMO PLAYER

Save the script. In this case, we'll use the name "Assign Parameter to Elements". Open the Dynamo Player, located in the Manage tab.



You will see the Dynamo Player interface, with a bunch of default scripts. Click on the folder icon with an arrow. A windows folder will pop up. Paste your new script in that folder and click the circle refresh arrow.



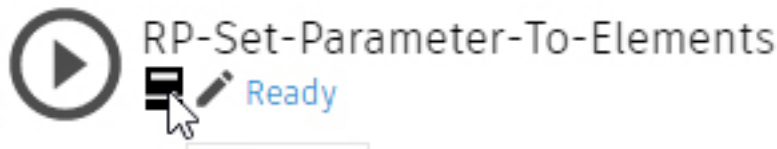


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## 11- TEST YOUR SCRIPT

Your script should now appear in the Dynamo Player. Click on the properties icon before running it:



Users can modify all inputs. We can use this script to modify any number parameter value from any category. In this case, let's try to alter the wall's base offset to 175mm. Watch out: the parameter names are cap-sensitive. Make sure you type them in exactly as they appear, or else you'll get a warning.

Select Model Elements :  
Select  
Count: 6 ...show more

CATEGORY :  
Walls

PARAMETER NAME :  
Base Offset

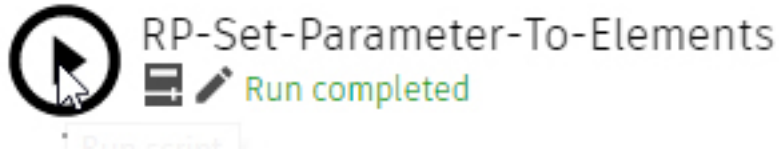
VALUE :  
175



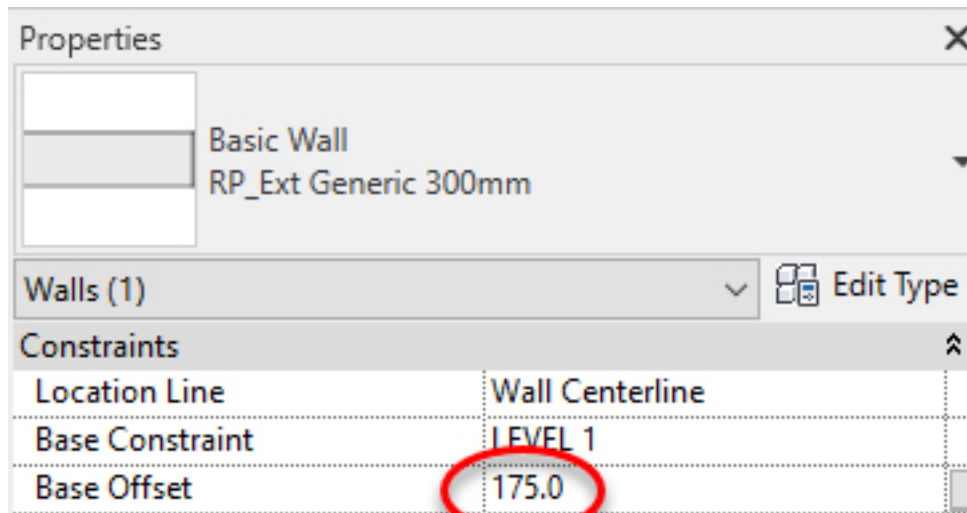
# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

Press on the **Play** button. If all goes well, it should say “Run completed”. If you get a warning, you’ll have to debug your script or ensure the inputs don’t have any mistakes.



To make sure the script worked, verify the walls on the models. The selected walls indeed have a Base Offset value of 175mm.





# PAMPHLETS

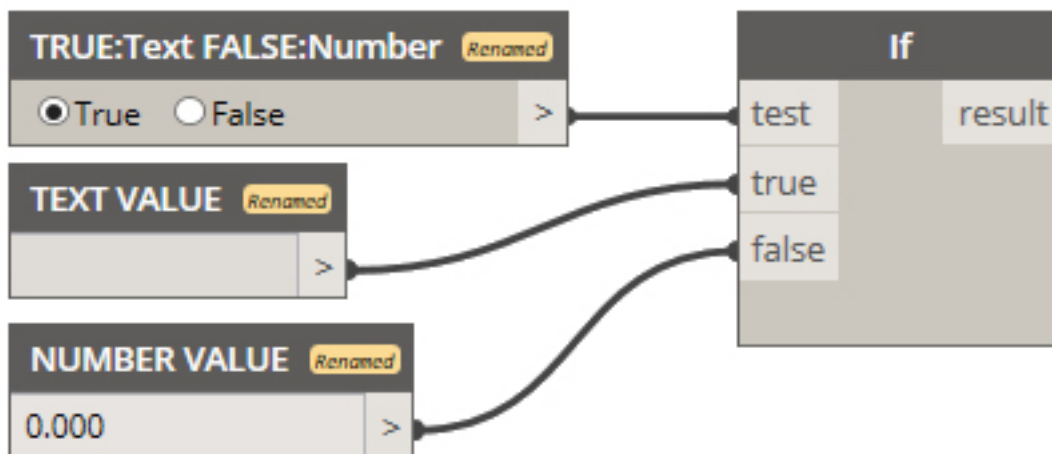
ISSUE #22 / FALL 2021 / DYNAMO

## 12- ADD A "STRING" PARAMETER OPTION

In its current state, this script only allows us to assign values to numerical parameters. What if you want to modify parameters with text values?

Let's add an option. Bring in a **Boolean** node and a **If** node. As you can see below, a boolean node is an input node with true/false values. Users will be able to pick an option in the Dynamo player. In the example below, we rename the node to give some instructions to users.

The **If** node will return a text value if true and a number value if false.



Plug the **If** node output to the **value** input of the SetParameter node. Make sure the **Boolean** node and the **String** node are set to Input.



# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

Save the script and open the Dynamo Player again. You should see this:

⋮

✓ Select Model Elements :  
  
Count: 12 ...show more

✓ CATEGORY :

✓ PARAMETER NAME :

✓ TRUE:Text FALSE:Number :  
 True

✓ TEXT VALUE :

✓ NUMBER VALUE :

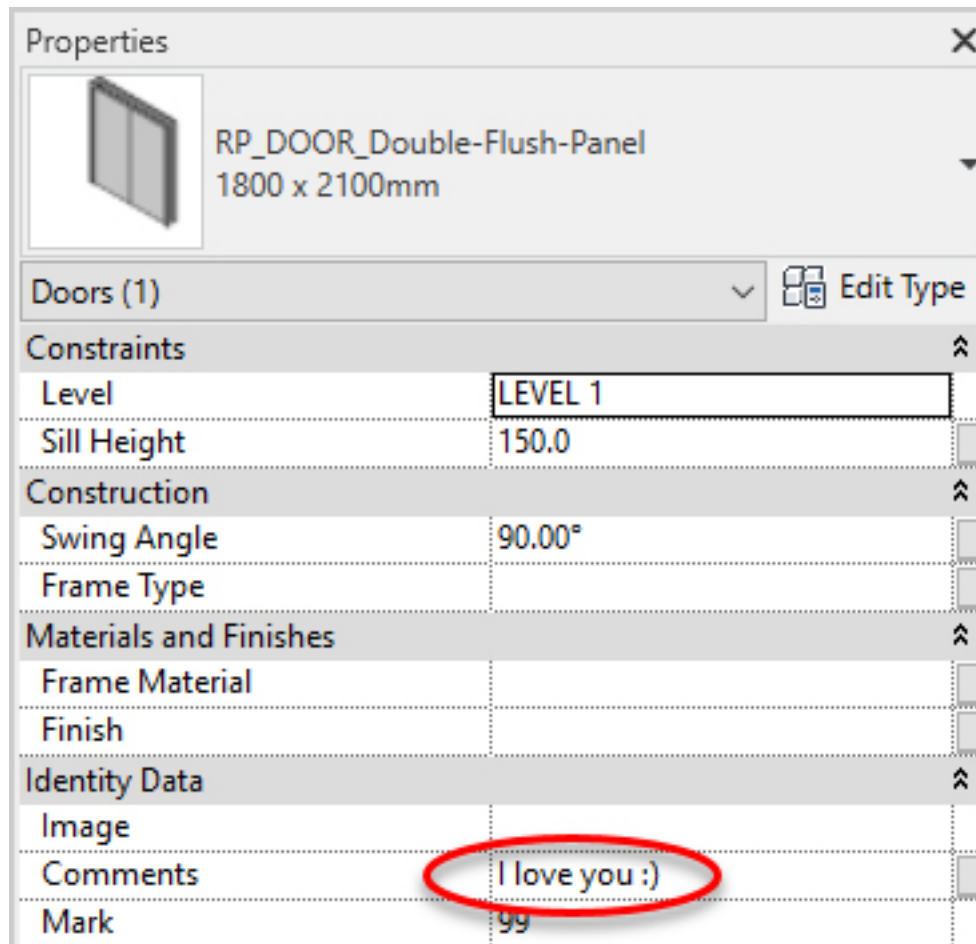
In this script, the NUMBER VALUE is ignored because we've set the switch to True. Only the TEXT VALUE is used.



# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

Have a look at the doors: they have the text parameter we've set in the Comments field. Love is in the air.



If you have paid attention, you've probably realized you could modify any category parameter using this script. Amazing, isn't it?



# PAMPHLETS

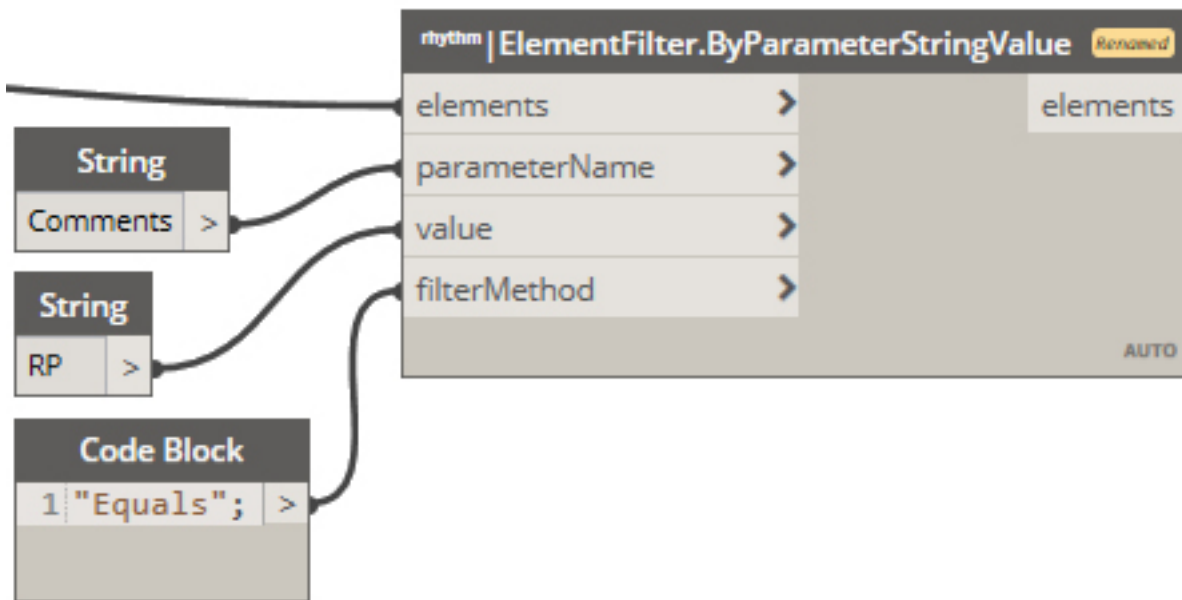
ISSUE #22 / FALL 2021 / DYNAMO

## 13- ADD PARAMETER FILTERING

In this case, we affect all selected elements of a specific category. What if you want to only affect elements with specific parameter values?

Let's bring in another node from the Rhythm package called **ElementFilter.ByParameterStringValue**.

In the example below, we use this node filter to only keep elements whose **Comments** parameter is equal to **RP**. The rest of the script would only apply to these specific elements.





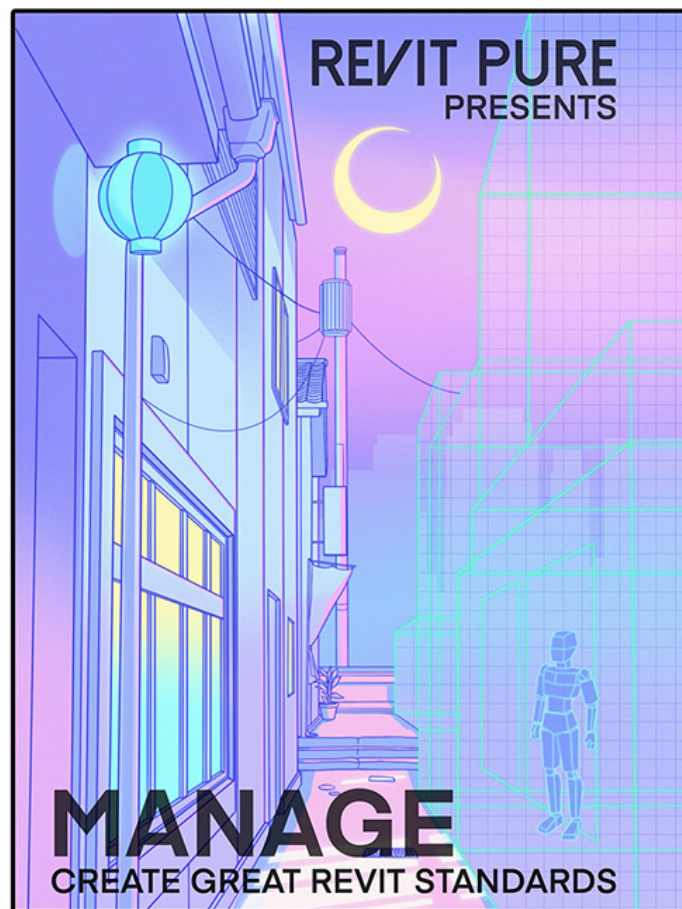


# PAMPHLETS

ISSUE #22 / FALL 2021 / DYNAMO

## UNLOCK THE REST

This PDF is an excerpt from our new MANAGE learning package for Revit. The full course contains even more Dynamo learning content. You will learn how to use Excel with Dynamo, you will set a sequence of numbers to elements' marks and you will learn how to make Dynamo more user friendly. [Click here](#) to learn more about the MANAGE course and get 15% off.





# PAMPHLETS

---

ISSUE #22 / FALL 2021 / DYNAMO

## THANKS FOR READING!

As always, send your thoughts to [nick@revitpure.com](mailto:nick@revitpure.com). I read and answer all emails. Let me know what theme you want explored for the next edition.