

# Dynamo for Grasshopper users

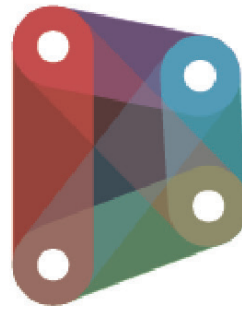


grasshopper

*/ˈgrɑːʃɒpə/*  
noun

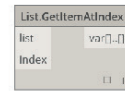


=



dynamo

*/ˈdaɪnəməʊ/*  
noun





## Introduction

A while back I took the plunge and tried to learn Dynamo. As a Grasshopper user, it was quite frustrating knowing the computational workflow required but not the equivalent commands in Dynamo. After a lot of research I developed this list of tips and translations to help Grasshopper users learn Dynamo. I hope that you find it useful in migrating your Grasshopper skills into Dynamo.

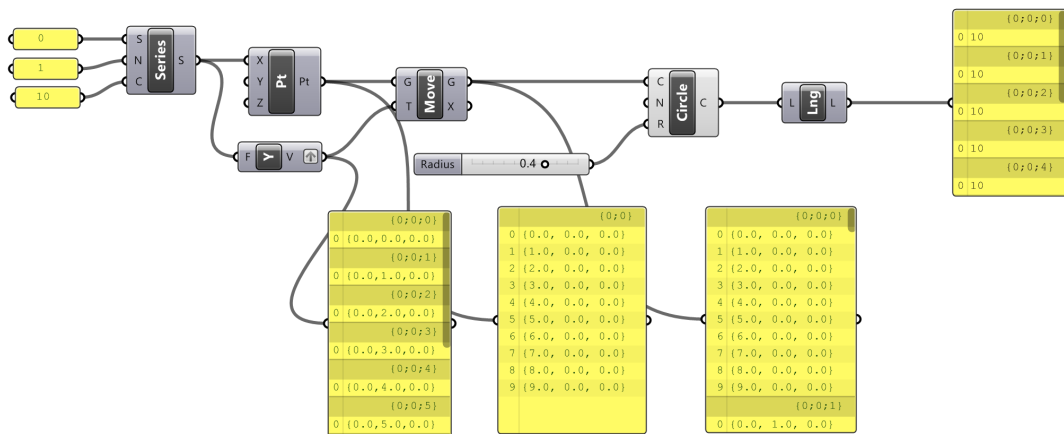
Paul Wintour



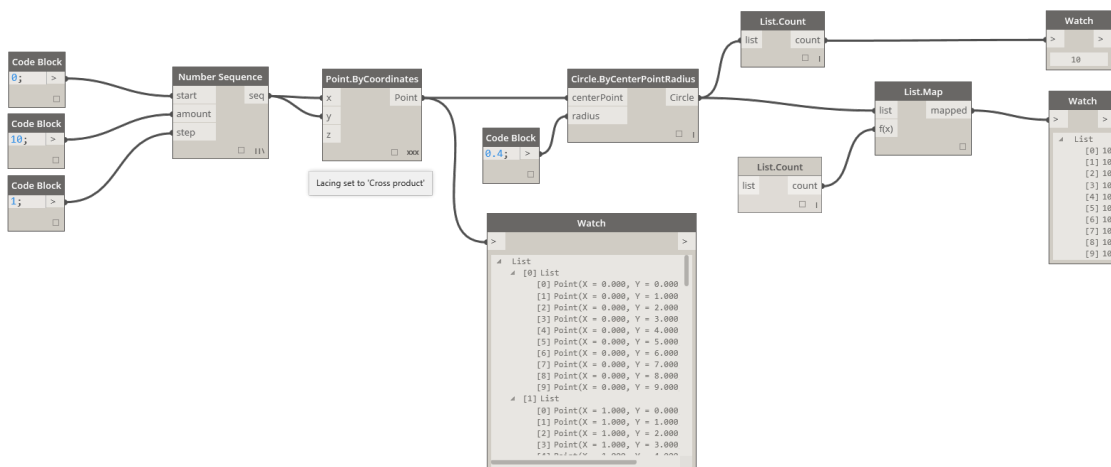
[www.parametricmonkey.com](http://www.parametricmonkey.com)

## Tips

- Probably the most difficult concept for Grasshopper users to grasp in Dynamo in how Dynamo handles nested lists. In Grasshopper you will be familiar with data trees and the need for them to match. Components such as Graft, Flatten, Simplify and Path Mapper become critical. The image below illustrates (the long way) how to create a simple 2-dimensional array of circles. Once one row is created, a series of numbers are grafted, creating multiple data paths. Feeding this into the circle component we get a list if lists.



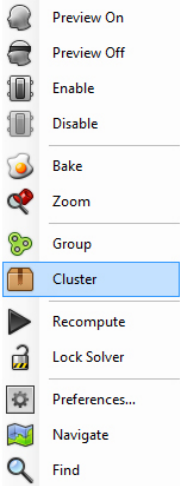
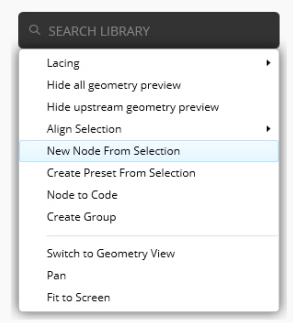

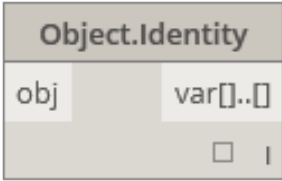
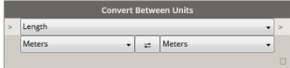


In Dynamo however, we can simplify this by using lacing when creating the points. By setting the lacing to 'cross product' we automatically get a 2-dimensional array of points. While certain nodes will work on a list of lists, others don't. For example, the generation of the circles works in pretty much the same way, no data tree modification is required. However, if we want to calculate the length of each nested list, we need to use the 'List.Map' node. This node repeats the function to the other data trees. Note how no input is connected to the List.Count node. The list of lists is fed directly into the List.Map node. This is fundamentally different to how Grasshopper processes data and can become incredibly confusing for Grasshopper users. But fear not, there are plenty of video tutorials on the topic, like [this one](#) from Autodesk.



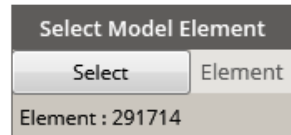
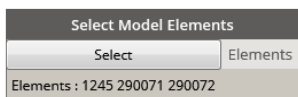

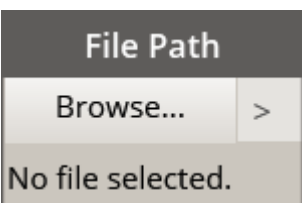

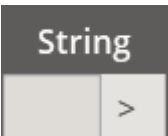
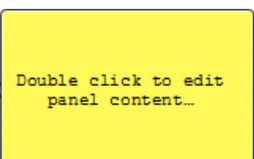
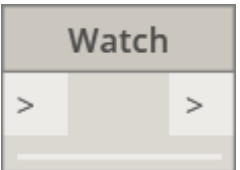

- [As of version 0.8](#), Dynamo is fundamentally unitless. This allows Dynamo to remain an abstract visual programming environment. Dynamo nodes that interact with Revit dimensions will reference the Revit project's units.
- Dynamo in sandbox mode doesn't have Revit nodes.




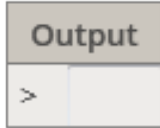
# Translation

## General:


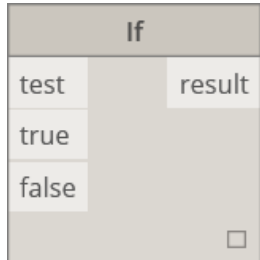
Grasshopper	Dynamo	Notes
<p>Cluster</p> 	<p>Custom Node</p> 	<p>Custom Nodes are constructed by nesting other nodes and custom nodes inside of a “Dynamo Custom Node,” which we can think of conceptually as a container. When this container node is executed in your graph, everything inside it will be executed to allow you to reuse and share a useful combination of nodes.</p>
<p>Data</p> 	<p>Object.Identity</p> 	<p>Returns what is parsed in, doing nothing.</p>
<p>N/A</p>	<p>Convert Between Units</p> 	<p>Converts between units of measure.</p>
<p>N/A</p>	<p>Element.Geometry</p> 	<p>Gets all of the geometry associated with this object.</p>
<p>N/A</p>	<p>Code Block</p> 	<p>Allows for DesignScript code to be authored directly. When using code blocks for numbers, you need to include all numbers . For example, you cannot use “.2”. It must be “0.2”. There is no difference for addition and concatenate, simply use ‘+’ and Dynamo will figure out the rest based on the input it receives. Check out <a href="#">this</a> blog post for more information.</p>

## Parameters:


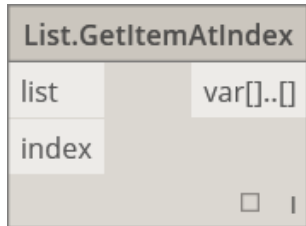
Grasshopper	Dynamo	Notes
<p>Set One Geometry</p> <hr/> <p>Set one Geometry</p> <p>Set Multiple Geometries</p> <p>Manage Geometry collection</p>	<p>Select Model Element</p> 	<p>Selects a model element from the Revit document. Usually combined with Element.Geometry.</p>
<p>Set Multiple Geometries</p> <hr/> <p>Set one Geometry</p> <p>Set Multiple Geometries</p> <p>Manage Geometry collection</p>	<p>Select Model Elements</p> 	<p>Selects multiple elements from the Revit document. Usually combined with Element.Geometry. Note to select multiple items you need to drag a window. Using tab select won't work.</p>
<p>File Path</p> 	<p>File Path</p> 	<p>Allows you to select a file on the system to get its filename. Always combined with the File.FromPath node.</p>
<p>Text / Panel</p> 	<p>String</p> 	<p>Creates a string.</p>
<p>Panel</p> 	<p>View.Watch</p> 	<p>Visualize the output of node.</p>
<p>Scribble</p> <p><b>Doubleclick Me!</b></p>	<p>Note</p> 	<p>Edit &gt; Create Note (Ctrl+W).</p>


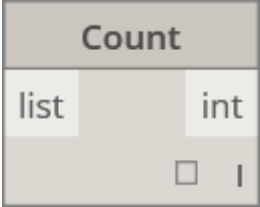
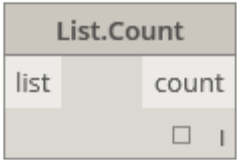

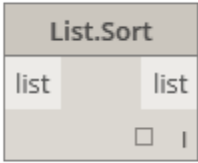
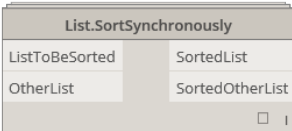


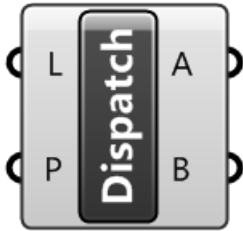
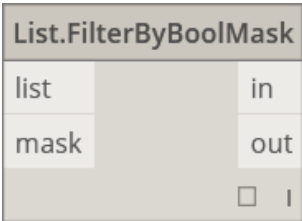
Cluster Input 	Input 	Note: Only available in the Custom Node environment.
Cluster Output 	Output 	Note: Only available in the Custom Node environment.

### Maths:


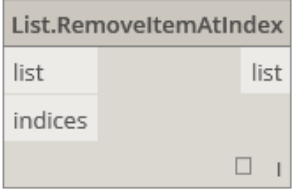

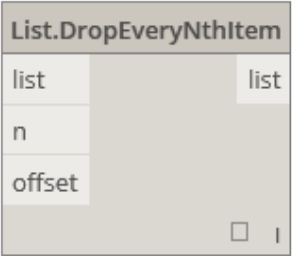

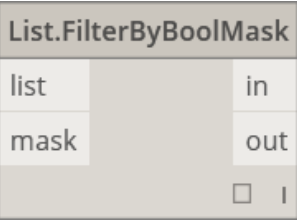
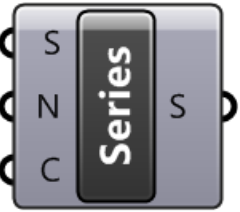
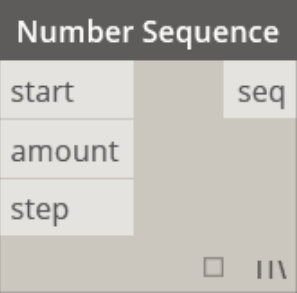
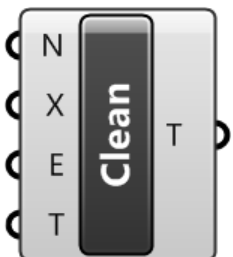
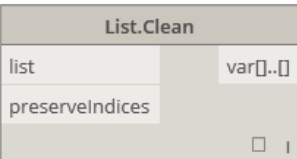
Grasshopper	Dynamo	Notes
Evaluate 	If 	Conditional statement. Checks the Boolean value of the test input. If the test input is true, the result outputs the true input, otherwise the result outputs the false input. Usually combined with Dispatch (Grasshopper) / List.FilterByBoolMask (Dynamo).


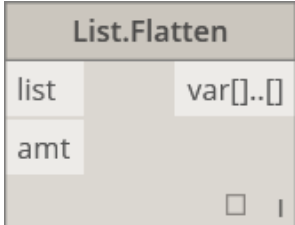



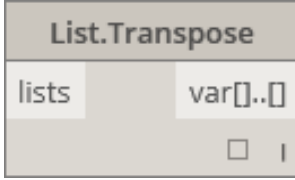

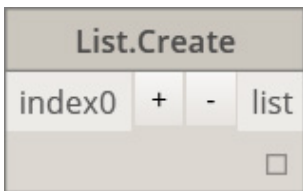
### Sets:

Grasshopper	Dynamo	Notes
List item 	List.GetItemAtIndex 	Gets an item from the given list that's located at the specified index.

<p>List Length</p> 	<p>Count</p>  <p>List.Count</p> 	<p>Returns number of items in the specified list. These nodes are identical. One of them should be retired.</p>
<p>Sort List</p> 	<p>List.Sort</p>  <p>List.SortSynchronously</p> 	<p>Sorts a list using the built-in natural ordering. For a more accurate interpretation of the Grasshopper component, use the List.SortSynchronously node found in the BVN Dynamo Package.</p>
<p>Reverse List</p> 	<p>List.Reverse</p> 	<p>Creates a new list containing the items of the given list but in reverse order.</p>
<p>Dispatch</p> 	<p>List.FilterByBoolMask</p> 	<p>Filters a sequence by looking up corresponding indices in a separate list of Booleans.</p>


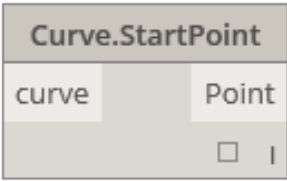
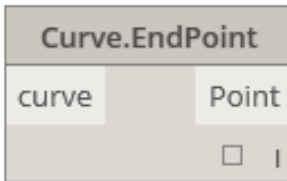

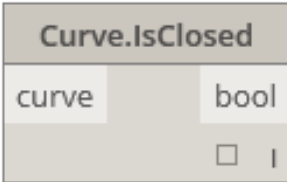
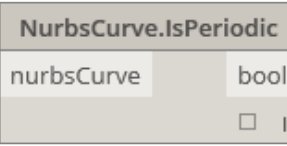

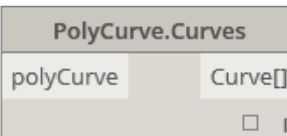


<p>Cull Index</p> 	<p>List.RemoveItemAtIndex</p> 	<p>Removes an item from the given list at the specified index.</p>
<p>Cull Nth</p> 	<p>List.DropEveryNthItem</p> 	<p>Removes items from the given list at indices that are multiples of the given value, after the given offset.</p>
<p>Cull Pattern</p> 	<p>List.FilterByBoolMask</p> 	<p>Filters a sequence by looking up corresponding indices in a separate list of Booleans.</p>
<p>Series</p> 	<p>Number Sequence</p> 	<p>Creates a sequence of numbers.</p>
<p>Clean Tree</p> 	<p>List.Clean</p> 	<p>Cleans data of nulls and empty lists from a given list of arbitrary dimensions.</p>


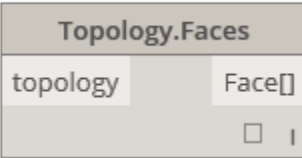
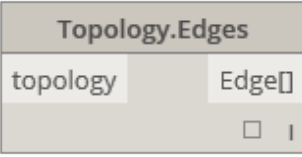
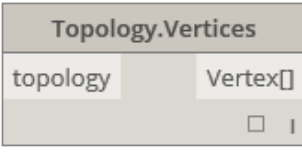

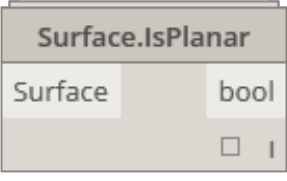

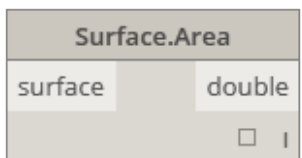
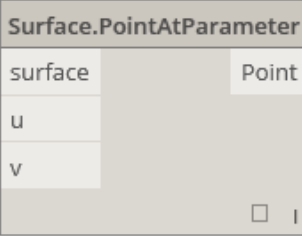
<p>Flatten Tree</p> 	<p>List.Flatten</p>  <p>Flatten</p> 	<p>Flattens a nested list of lists by a certain amount.</p> <p>Returns the flattened 1D list of the multi-dimensional input list.</p>
<p>Simplify Tree</p> 	<p>N/A</p>	
<p>Flip matrix</p> 	<p>List.Transpose</p> 	<p>Flips columns and rows.</p>
<p>Merge</p> 	<p>List.Create</p> 	<p>Makes a new list out of the given inputs.</p>


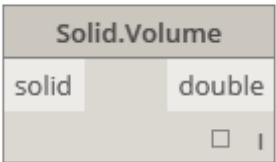
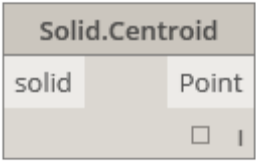
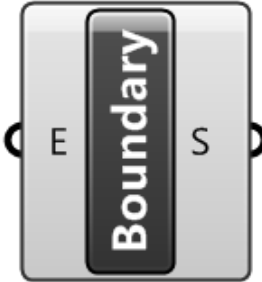
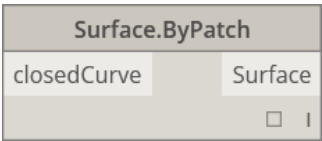


## Curve:

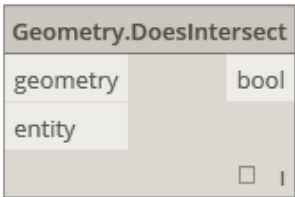
Grasshopper	Dynamo	Notes
<p>End Points</p> 	<p>Curve.StartPoint</p>  <p>Curve.EndPoint</p> 	<p>You will need to use two nodes to get the start and end points of a curve.</p>
<p>Closed</p> 	<p>IsClosed</p>  <p>NurbsCurve.IsPeriodic</p> 	<p>Depending on the input, either use IsClosed for polycurves or NurbsCurve.IsPeriodic</p>
<p>Explode</p> 	<p>PolyCurve.Curves</p> 	<p>Returns curves of the polycurve. Geometry. Explode can also be used. Since Revit doesn't have polycurves, you will need to extract the subcurves in order to create detail or model lines.</p>

## Surface:


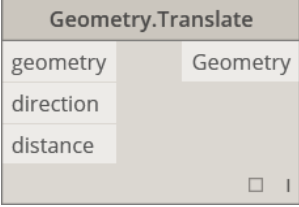
Grasshopper	Dynamo	Notes
<p>Deconstruct Brep</p> 	<p>Topology.Faces</p>  <p>Topology.Edges</p>  <p>Topology.Vertices</p> 	<p>'Topology' in Dynamo is a parent category for Surfaces, Polysurfaces, and Solids. To extract the faces, edges and vertices, you will need to use 3 separate nodes. An alternative methodology is to use 'Geometry.Explode' and then either: Face.SurfaceGeometry; Face.Edges; or Face.Vertices.</p>
<p>Is Planar</p> 	<p>Surface.IsPlanar</p> 	<p>There is no out of the box node to test if a surface is planar. You will need to use the Surface.IsPlanar node from the Clockwork package.</p>
<p>Area</p> 	<p>Surface.Area</p>  <p>Surface.PointAtParameter</p> 	<p>To get the area and centroid of a surface you will need to use two nodes. To get the centroid to the surface enter 0.5 for the U and V inputs of the Surface.PointAtParameter node.</p>

<p>Volume</p> 	<p>Solid.Volume</p>  <p>Solid.Centroid</p> 	<p>To get the volume and centroid of a solid you will need to use two nodes.</p>
<p>Boundary Surfaces</p> 	<p>Surface.ByPatch</p> 	<p>Create a surface by filling in the interior of a closed boundary defined by input curves.</p>

**Intersect:**

Grasshopper	Dynamo	Notes
<p>Various</p>	<p>Geometry.DoesIntersect</p> 	<p>Determines if another geometry object intersects with this one.</p>

## Transform:

Grasshopper	Dynamo	Notes
<p data-bbox="140 392 210 421">Move</p>  <p>The icon for the Move component in Grasshopper. It is a rectangular box with rounded corners. In the center, there is a dark vertical bar with the word "Move" written in white. On either side of this bar, there are two ports. The top ports are labeled "G" and the bottom ports are labeled "T" and "X".</p>	<p data-bbox="491 392 727 421">Geometry.Translate</p>  <p>The Dynamo component for Geometry.Translate. It is a rectangular box with a title bar that says "Geometry.Translate". Below the title bar, there are three input ports: "geometry" (with a dropdown menu showing "Geometry"), "direction", and "distance". There is a small square icon and a vertical line at the bottom right of the component.</p>	<p data-bbox="823 392 1474 450">Translates any geometry type by the given distance in the given direction.</p>