

Dynamo for BIM Managers

aka Managing Dynamo and Dynamo for Managing

Mostafa El Ayoubi

Design Technology Expert | Data Shapes

John Pierson

Design Technology n00b | Parallax Team

Aaron Maller

Director | Parallax Team

Sol Amour

Product Manager : Dynamo | Autodesk

Jostein Olsen

Good Monkey | Bad Monkeys

DYNAMO WORKSHOP



preflight

Package Deployment Options

Name	Date modified
DW331545-L-LoadPackages.v1.0.0.exe	11/15/2019 3:20 PM


```
; Script generated by the Inno Setup Script Wizard.
; SEE THE DOCUMENTATION FOR DETAILS ON CREATING INNO SETUP SCRIPT FILES!

#define MyAppName "AU2019 - Load Packages for DW331545-L Dynamo for BIM Managers"
#define MyAppVersion "1.0.0"
#define MyAppPublisher "Autodesk University"
#define MyAppURL "https://github.com/DynamoCD/DynamoWorkshops"

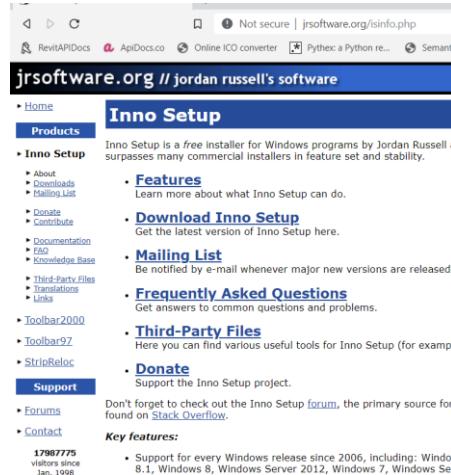
#define DynamoPackageFolder22 "{userappdata}\Dynamo\Revit\2.2\packages"
#define SamplesFolder22 "C:\ProgramData\Autodesk\RVT 2020\Dynamic\samples\en-US\Templates"

[Setup]
; NOTE: The value of AppId uniquely identifies this application.
; Do not use the same AppId value in installers for other applications.
;(To change the GUID, click Tools | Generate GUID inside the IDE.)
;(john) I went ahead and signed this addin for the actual example. For future reference, you will need to do this yourself.
SignTool=prismalign
SignToolPrismalign
AppId={F0A9E9C7C2-4557-9252-94CE30900D75}
AppName={#MyAppName}
AppVersion={#MyAppVersion}
AppPublisher={#MyAppPublisher}
AppURL={#MyAppURL}
AppSupportURL={#MyAppURL}
AppSupportPhone=(702) 690-7860
AppSupportEmail=_resources\AU.ico
CreateAppDir=no
OutputBaseFilename=DW331545-L-LoadPackages.v{#MyAppVersion}
SetupIconFile=_resources\AU.ico
UninstallIconFile=_resources\AU.ico
Uninstallable=no
SolidCompression=yes

[Languages]
Name: "english"; MessagesFile: "compiler:Default.isl"

[Components]
Name: "PackageInstall"; Description: "AU2019 - Load Packages for DW331545-L Dynamo for BIM Managers"; Types: full;

[Files]
; this embeds the Dynamo packages in the installer.
Source: "\Bundle\Package\*"; DestDir: {#DynamoPackageFolder22}; Flags: recursesubdirs; Components: PackageInstall
Source: "\Bundle\_template.dyn"; DestDir: {#SamplesFolder22}; Components: PackageInstall
```



The screenshot shows a web browser displaying the jrssoftware.org website. The URL bar shows "Not secure | jrssoftware.org/sinfo.php". The main content area has a blue header bar with the text "jrssoftware.org // jordan russell's software". Below this is a navigation menu with "Inno Setup" as the active item. The "Inno Setup" section contains several links: "About", "Downloads", "Mailing List", "Donate", "Contribute", "Documentation", "FAQ", "Knowledge Base", "Third-Party Files", "Translations", and "Links". It also includes sections for "Features", "Download Inno Setup", "Mailing List", "Frequently Asked Questions", "Third-Party Files", "Support", "Forums", and "Contact". A sidebar on the right displays statistics: "17907775 visitors since Jan. 1998".

A photograph of a utility pole from a low angle, looking up. The pole is made of light-colored wood and has several metal brackets and insulators holding multiple sets of power lines. The lines are dark and crisscross against a bright, clear blue sky. In the center of the image, there is a solid black rectangular overlay containing the text.

unraveling dynamo

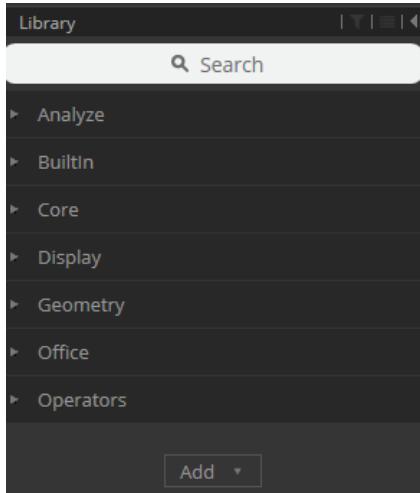
Dynamo Review



Dynamo 1.3.4

- Support for Revit 2017-2019
- XML-Based.
- Pretty darn stable!
- Being phased out with Revit 2020

Version Differences - 1.3.4



```
1 <Workspace Version="1.3.2.2480" X="-419.949656255947" Y="-
2 <NamespaceResolutionMap />
3 <Elements>
4   <Dynamo.Graph.Nodes.ZeroTouch.DSFunction guid="75531b0
5     <PortInfo index="0" default="False" />
6     <PortInfo index="1" default="False" />
7     <PortInfo index="2" default="False" />
8     <PortInfo index="3" default="False" />
9   </Dynamo.Graph.Nodes.ZeroTouch.DSFunction>
10  <Dynamo.Graph.Nodes.CustomNodes.Function guid="9dfd34a
11    <PortInfo index="0" default="True" />
12    <ID value="e620b7b6-640d-435e-982a-15fc1eeb8bdb" />
13    <Name value="Collector_PlacedRooms" />
14    <Description value="This will collect placed rooms i
15      <Inputs>
16        <Input value="toggle" />
17      </Inputs>
18      <Outputs>
19        <Output value="rooms" />
20      </Outputs>
21    </Dynamo.Graph.Nodes.CustomNodes.Function>
22    <CoreNodeModels.Input.BoolSelector guid="e2c2c8d3-50a2
23    <System.Boolean>False</System.Boolean>
```

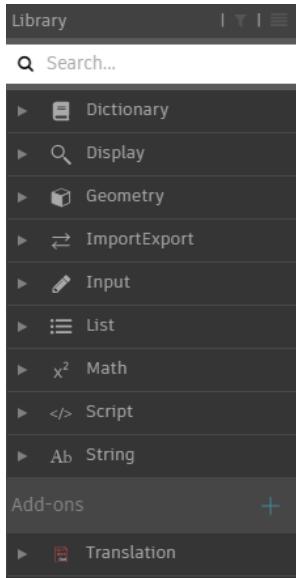
- Same library for several years.
- Packages bundled within the library categories
- Node in Dynamo 1.3.3

- XML-Based file format
- Note the "<>"

Dynamo 2.0.x

- Released with Revit 2018 and offers compatibility for 2017-2019
- JSON based approach
- Migration strategies are difficult resulting in slower adoption (IMO)
- Not backwards compatible with 1.3.x
- Enabling of extensions via package manager

Version Differences - 2.0.x



- New Appearance
- Add-ons category for custom packages
- Node in Dynamo 2.0.x
- Auto-Lacing by default

```
1 [
2   "Uuid": "652a7978-e658-4a67-91c3-fcb9a9875e9f",
3   "IsCustomNode": false,
4   "Description": null,
5   "Name": "dynamo2",
6   "ElementResolver": {
7     "ResolutionMap": {}
8   },
9   "Inputs": [],
10  "Outputs": [],
11  "Nodes": [
12    {
13      "ConcreteType": "Dynamo.Graph.Nodes.ZeroTouch.DSFunction", "DynamoCore",
14      "NodeType": "FunctionNode",
15      "FunctionSignature": "+@var[...][],var[],[]",
16      "Id": "cbe31123ed7e437838028bdfbdd2016",
17      "Inputs": [
18        {
19          "Id": "44b444ee5baa40160c9cfb71044f69f7",
20          "Name": "x",
21          "Description": "x value.\n\\nvar[...][]",
22          "UsingDefaultValue": false,
23          "Level": 2,
24          "Uselevels": false,
25          "KeepListStructure": false
26        },
27        {
28          "Id": "a333e56ff9e74e108c01c83a26072ab5",
29          "Name": "y"
30        }
31      ]
32    }
33  ]
34]
```

- JSON-Based file format
- Note the "{}"

Dynamo 2.1.0

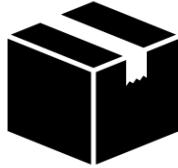
- Released with Revit 2020 and only available on Revit 2020 as integrated product.
- Remains compatible with other 2.0.x versions per [semantic versioning](#). *This means you have access to 2017-2020 compatibility.*
- Search improvements, overall more stable.

Future of Dynamo

- Dynamo is open source, so staying up to date is relatively easy if you know where to look.
- GitHub
- Blog/Forum
- Tweetz

Making Dynamo better

with Packages



Install these (**BOLD** at a minimum or you are doing Dynamo wrong)

Revit Related

- Lunchbox
- **Archi-lab.net**
- **Clockwork**
- **Spring nodes**
- Rhythm
- **Data-Shapes**
- Designtech
- bimorph
- Mandrill
- Bumblebee
- Bang!

More Awesomeness

- Ladybug
- Dynashape
- Firefly
- Mantis Shrimp
- Rhynamo



*Is your favorite package missing from this list. Let me know! (I reuse this slide all the time and it may need updated)

Useful View Extensions



DynaMonito

- DynaMonito - Andreas Dieckman
- <https://github.com/andydandy74/Monito>
- Top Features :
 - Fixing Groups
 - Finding nodes
 - Templating
 - Managing Inputs

Useful View Extensions



- DesignTech.io View Extension - Mark Thorley
- <https://github.com/MarkThorley/designtechViewExtension>
- Top Features:
 - Quick Access to Favorite Nodes
 - Quick Counts of Nodes/Wires

Useful View Extensions



- Monocle - by some guy named john pierson
- <https://www.parallaxteam.com/monocle-for-dynamo-released/>
- Top Features :
 - Documenting nodes
 - Clean Dynamo Graphs
 - Being Memetastic

Useful View Extensions



- IRIS for Dynamo - Eckart Schwerdtfeger
- <http://iris.parametrised.net/>
- Top Features :
 - Color coding beyond grouping
 - Serialization of data in file via node

Useful View Extensions



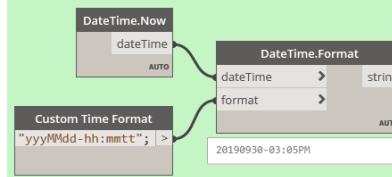
- The-Saurus for Dynamo - Whole Team
- <https://github.com/mitevpi/thesaurus>
- Top Features :
 - Predictive Text for Nodes
 - Best Logo
 - Working towards #AutodeskBuyout

Strategy: Dynamo Graph Run Tracking with Nodes

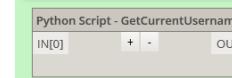
Information

MSDN Format Strings for Date and Time
<https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time>

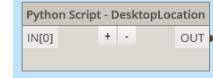
Gets the current time and formats it to our liking



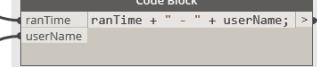
Python script to get the current windows user's login name
this being a python script eliminates package dependencies



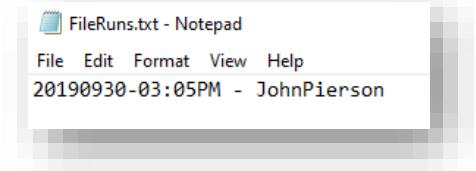
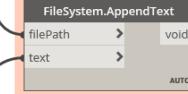
Write to desktop for now, can be another location



Join the run time and username



Write the file



Strategy: Dynamo Graph Run Tracking with Python

LANDING PAGE

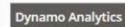
OFFICE: <Office>

AUTHOR NAME: <Author Name>

AUTHOR EMAIL: <Author Email>

KEYWORDS: <Keyword 1, Keyword 2, Keywords N>

This Python Script creates a CSV file to track the usage of the Dynamo file.



GET

Get parameter values from Host objects

SET

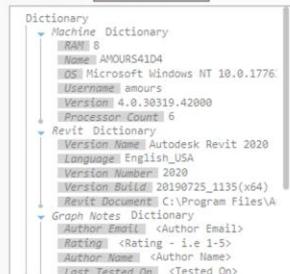
Set parameter values of Host objects

INPUT

Object creation in Host | Final output

FUNCTION

Data manipulation on dynamo objects



The screenshot shows a Microsoft Excel spreadsheet titled 'DynamoTracker_amours'. The table has columns L, M, N, O, P, and Q. Column L contains 'Graph: Name' and 'Graph: Path' entries. Column M contains 'Y:\OneDrive - autodesk\01 AEC Generative\Autodesk\dyn'. Column N contains 'Graph: Last Saved' with dates like '6/11/2019 3:25:38 PM'. Column O contains 'Graph: Node Count' with values like '3'. Column P contains 'Graph: Wire Count' with values like '0'. Row 6 is a blank row.

	L	M	N	O	P	Q
1	Graph: Name	Graph: Path	Graph: Last Saved	Graph: Node Count	Graph: Wire Count	Info
2	DynamoAnalyticsNode.dyn	Y:\OneDrive - autodesk\01 AEC Generative\Autodesk\dyn\AU 2019\DynamicAnalyticsNode.dyn	6/11/2019 3:25:38 PM	3	0	0
3	t	DynamoAnalyticsNode.dyn	Y:\OneDrive - autodesk\01 AEC Generative\Autodesk\dyn\AU 2019\DynamicAnalyticsNode.dyn	14/11/2019 11:30:24 AM	3	0
4	t	DynamoAnalyticsNode.dyn	Y:\OneDrive - autodesk\01 AEC Generative\Autodesk\dyn\AU 2019\DynamicAnalyticsNode.dyn	14/11/2019 11:30:24 AM	3	0
5	t	DynamoAnalyticsNode.dyn	Y:\OneDrive - autodesk\01 AEC Generative\Autodesk\dyn\AU 2019\DynamicAnalyticsNode.dyn	14/11/2019 11:30:24 AM	3	0
6						

Strategy: Dynamo Graph Run Tracking with Binoculars

The screenshot displays the Binoculars application interface, which integrates with Google Sheets and Google Data Studio.

Top Left: The Binoculars application window shows a menu bar with File, Edit, View, Packages, Settings, Help, and a central workspace with a "Binoculars" icon and a search bar.

Top Right: A summary dashboard titled "Dynamo Use Summary" provides real-time metrics: Active Users (100), Active Graphs (28), and Runs (781). It also includes a "Data From BINOCULARS" section.

Middle Left: A Google Sheets window titled "AU2019_Binoculars" is shown, displaying a table of data with columns for Username, Computer, IP, Geolocation, City, Country, Dynamo Version, Revit Version, Graph Name, Run Time, and Unique ID.

Bottom Right: A chart titled "Which graphs are used the most?" shows the distribution of runs across various graph names, with a legend indicating the count of runs.

Bottom Center: A status bar at the bottom of the dashboard indicates "Runs" (27) and "Runs (previous 30 days)" (245).

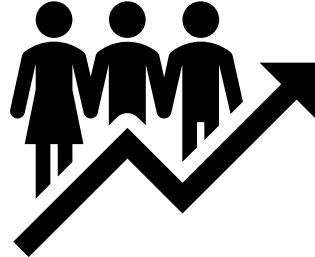
Bottom Right: A timeline or stats panel shows the period from Mar 7, 2019 - Apr 5, 2019.

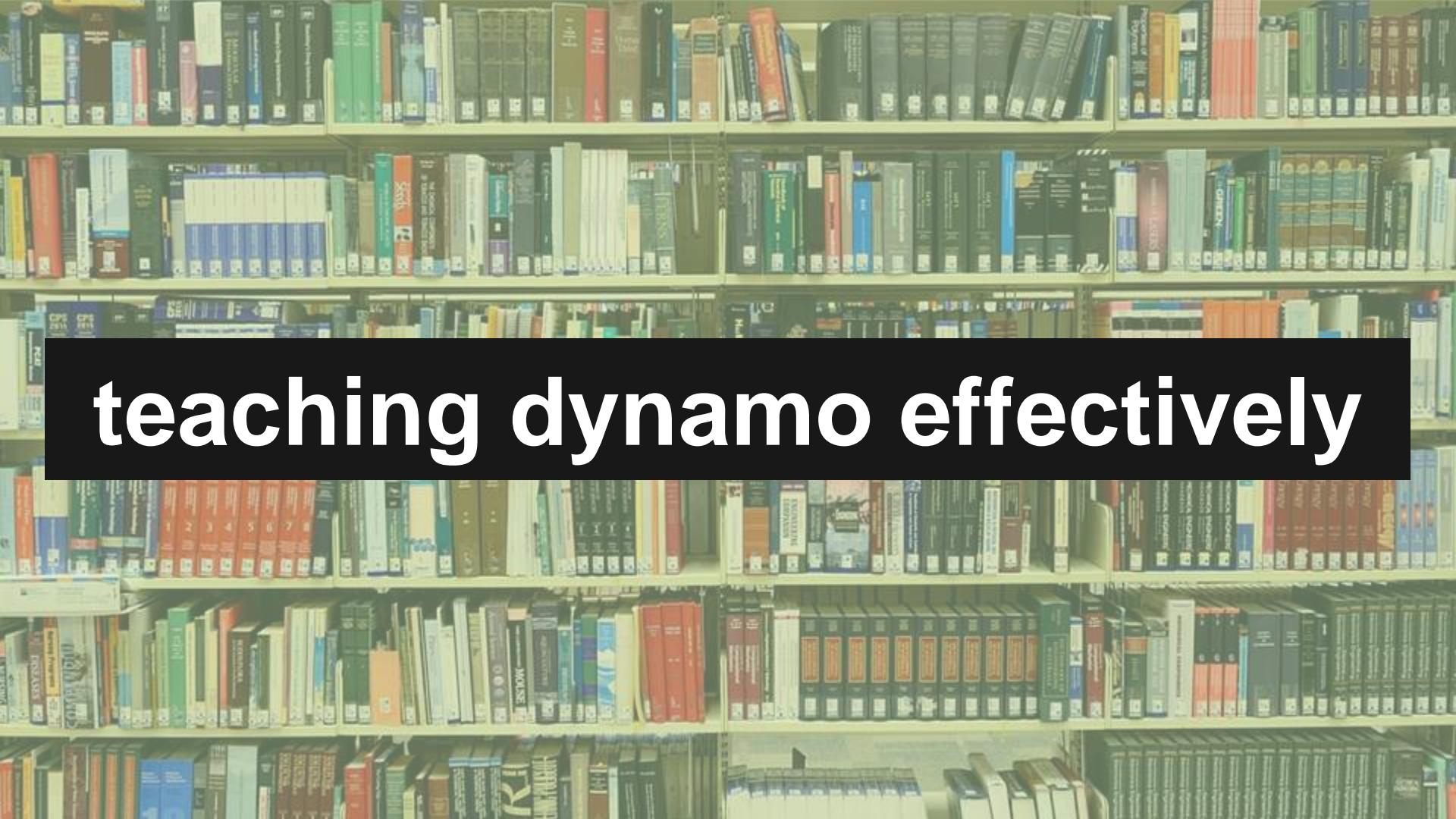
<https://github.com/teamtreedyn/Binoculars>

Dynamo Infrastructure



Element Binding?



The background of the slide features a photograph of a library shelf filled with books. The shelves are white, and the books are arranged in a dense, organized manner. The spines of the books are visible, showing various colors and titles, though they are not clearly legible.

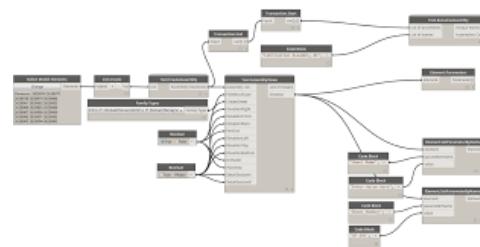
teaching dynamo effectively

Teaching Others



The Challenges

A very unusual way to interact with a design



Create a good atmosphere around it

Make a connection with real life use cases

**Remember, we were
all new Dynamo users
once.**

Lay the Foundations



Concept 1

Dynamo is a Visual Programming Environment

Programming

A set of instructions
telling a computer how to
perform a task or series
of tasks using one of
many **coding languages**

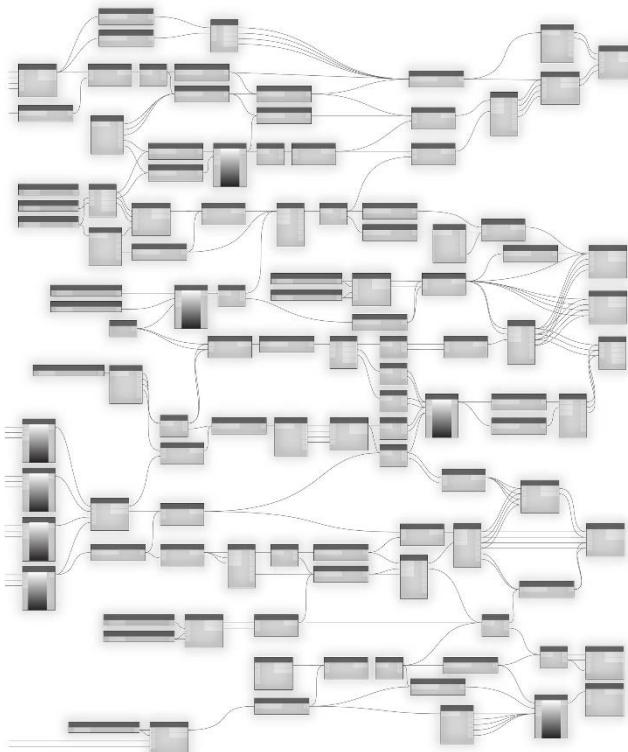
```

4 import clr
5
6 clr.AddReference('System.Windows.Forms')
7 clr.AddReference('System.Drawing')
8
9 from System.Drawing import Point, Size, Graphics, Ellipse, Image, Font, FontStyle, Icon, Color, Region, Rectangle
10 from System.Windows.Forms import Application, DockStyle, Button, Form, Label, Trackbar, ToolTip, ColumnHeader, TextBox, CheckBox, FolderBrowserDialog, OpenFileDialog, DialogResult, ComboBox, FormBorderStyle, FormStartPosition, ListView, ListView,
11 ListViewStyle, SortOrder, Panel, ImageLayout, GroupBox, RadioButton, BorderStyle, PictureBox, PictureboxSizeMode, LinkLabel, CheckBoxState, ColumnHeaderStyle, ImageList, VScrollBar
12
13 from System.Collections.Generic import List as list
14
15 from System.Windows.Forms import View as vlist
16
17 clr.AddReference('System')
18
19 from System import IntPtr
20
21 from System.ComponentModel.Design import Container
22
23
24 import sys
25 pyt_path = r'C:\Program Files (x86)\IronPython 2.7\lib'
26 sys.path.append(pyt_path)
27
28 os = None
29 import webbrowser
30 import unicodedata
31
32
33 clr.AddReference('RevitAPI')
34
35 from Autodesk.Revit.UI import Selection, TaskDialog
36 from Autodesk.Revit.Selection import SelectionFilter
37
38 importcolorselection = 0
39
40 try:
41     from Autodesk.Revit.UI import ColorSelectionDialog
42 except:
43     importcolorselection = 1
44
45
46 clr.AddReference('RevitNodes')
47 Import Revit
48 Import DocumentSet
49 Import Geometry
50 Import XYZ
51 Import ExtensionElement
52 Import GeometryConversion
53
54
55 clr.AddReference('RevitServices')
56
57 from RevitServices.Persistence import DocumentManager
58 doc = DocumentManager.Instance.CurrentDBDocument
59 uidoc = DocumentManager.Instance.CurrentUIApplication.ActiveUIDocument
60
61
62 clr.AddReference('System')
63
64 from Autodesk.Revit.DB import FilteredElementCollector, View, ViewType, ElementId, FamilyInstance
65 dbviews = [v for v in FilteredElementCollector(doc).OfClass(View).ToElements() if (v.ViewType == ViewType.FloorPlan or v.ViewType == ViewType.CeilingPlan or v.ViewType == ViewType.Section or v.ViewType == ViewType.Elevation or v.ViewType ==
66 ViewType.ThreeD)]
67
68
69
70
71 from System.Reflection import Assembly
72 import sal.structureElementTree as et
73
74 class selectorElement(SelectionFilter):
75     def __init__(self, category):
76         self.category = category
77
78     def GetElementFilter(self):
79         return self.category

```

Visual Programming

A set of instructions
telling a computer how to
perform a task or series
of tasks using a **nodal**
interface

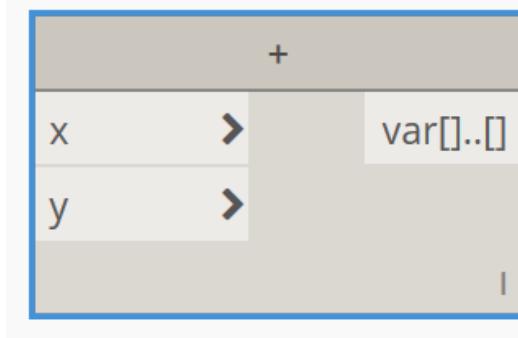


Code

```
def addition(a,b):  
    sum = a + b  
    return sum  
  
z = addition(x,y)  
print(z)
```

Python

Node

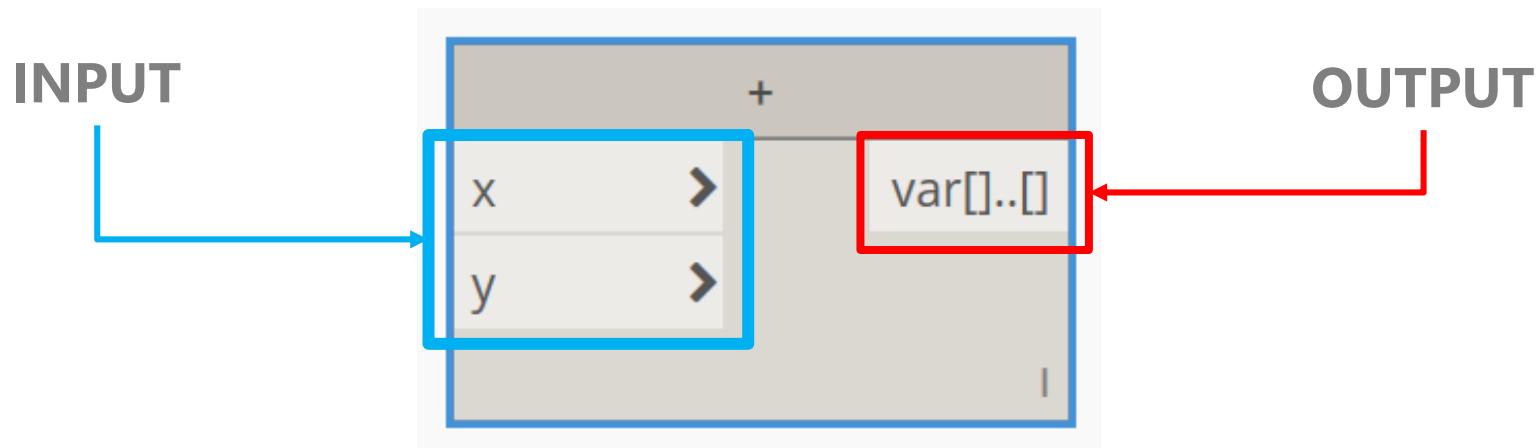


Dynamo

Concept 2

Understand the Data Flow

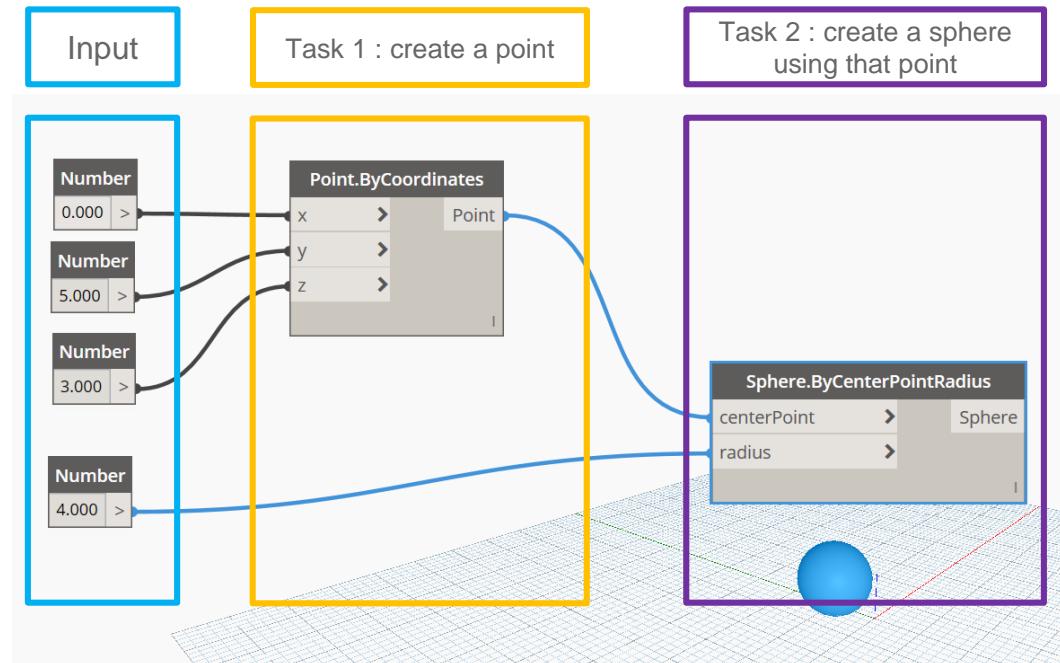
Anatomy of a Dynamo node



Anatomy of a Dynamo node



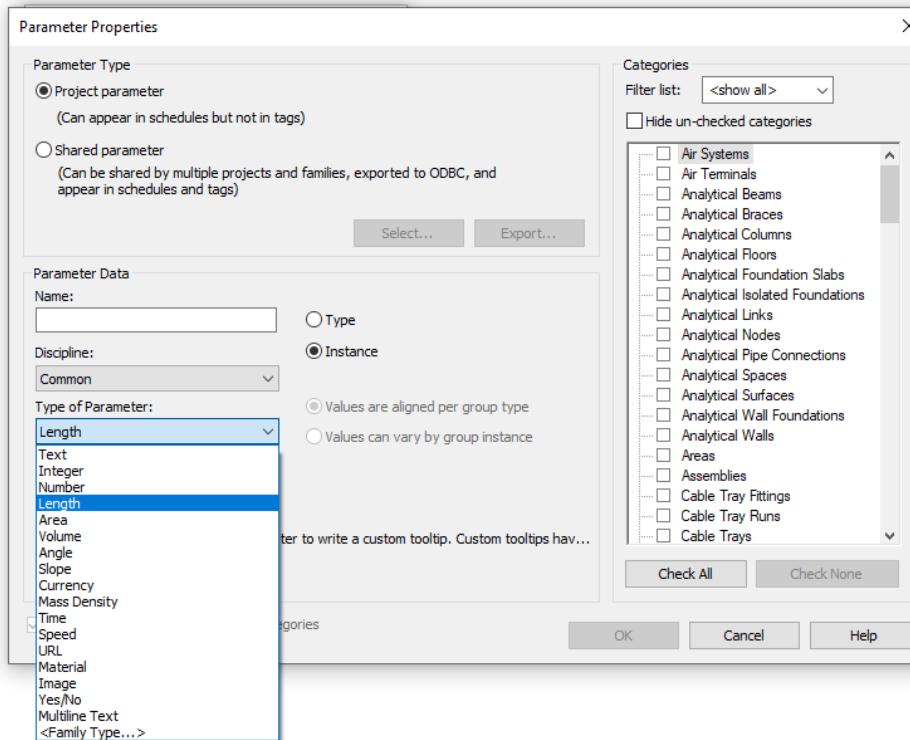
Anatomy of a Dynamo Script



Concept 3

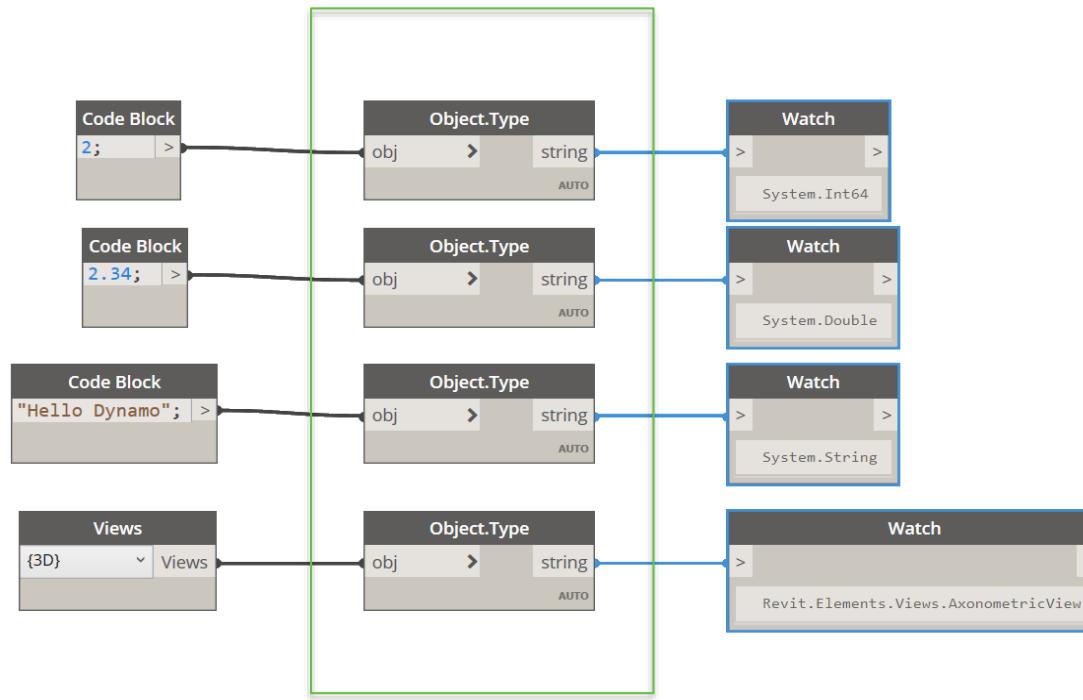
Understand the Data Types

Data Types



Data Types

If you want to figure the type of an element in Dynamo,
there is a super useful node called “Object.Type”



Data Types

In general:

Integers: 1 , 2 , 2938

Float or Double: 2.367 , 653.87

String: Hello World

Boolean: True, False

Lists: [banana, apple, pineapple]

Geometry elements: lines, solids , meshes

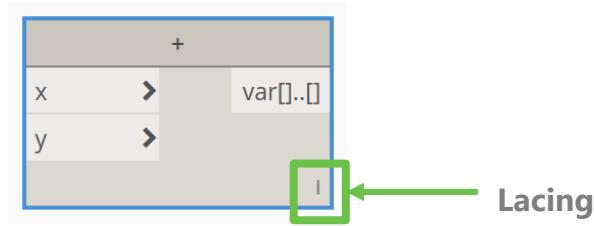
Working With Dynamo for Revit:

Revit Elements: walls, floors, rooms etc.

Some More advanced Concepts



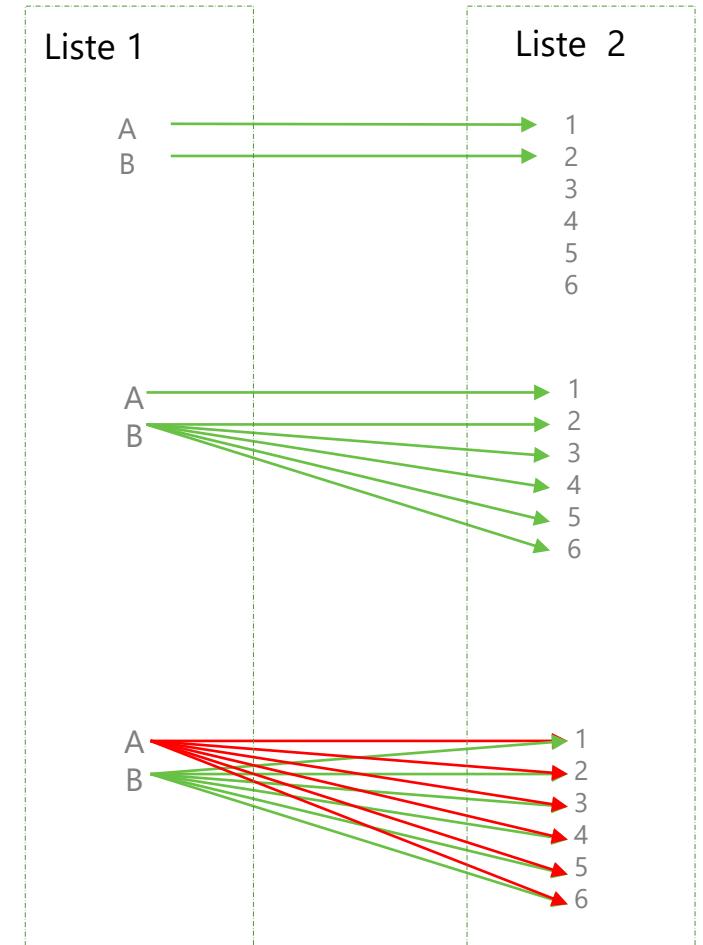
Lacing



Shortest

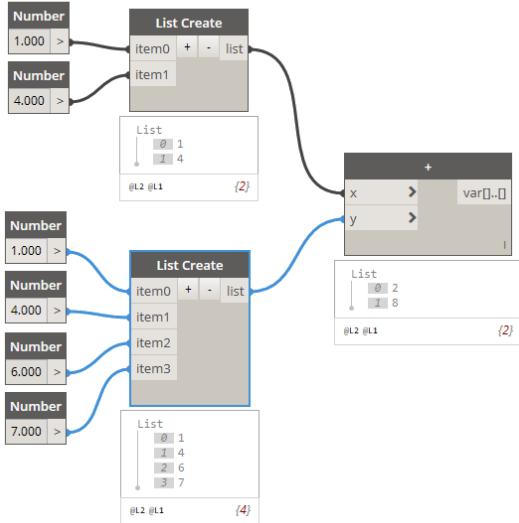
Longest

Cross Product

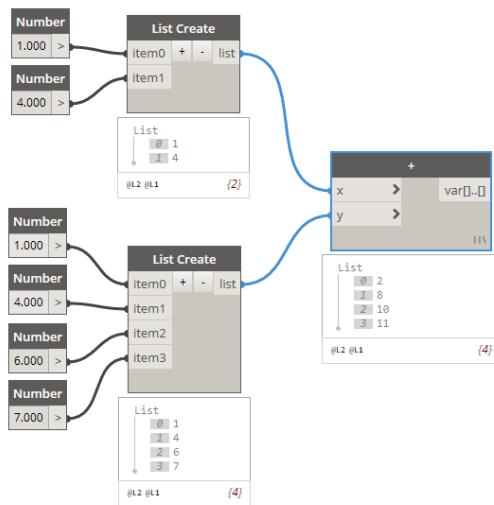


Lacing

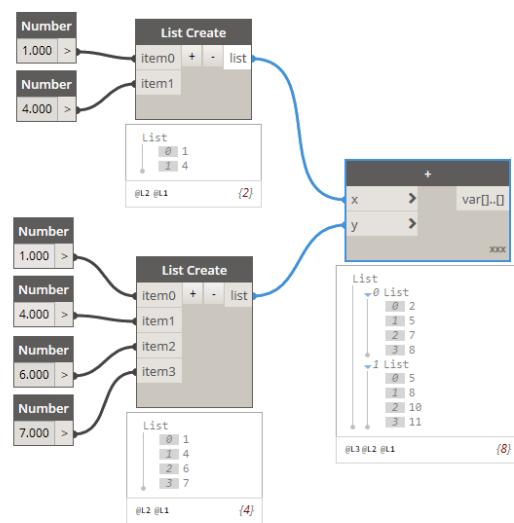
Shortest



Longest

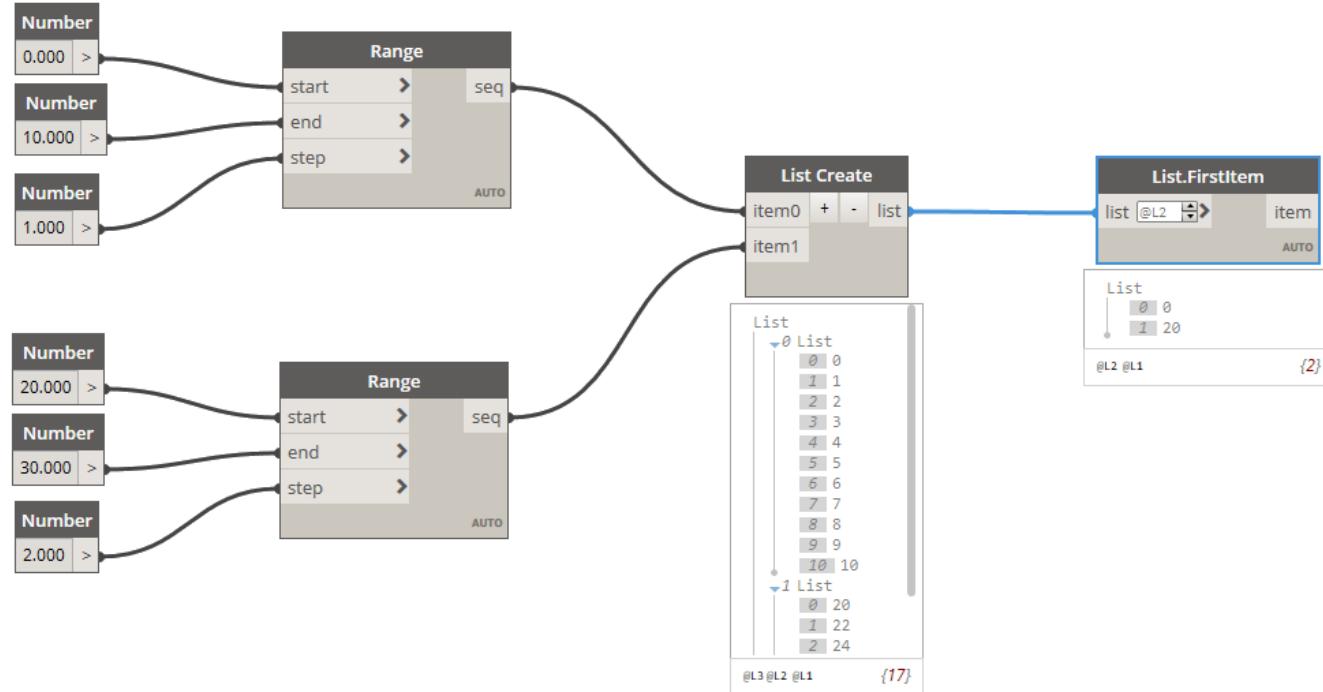


Cross Product



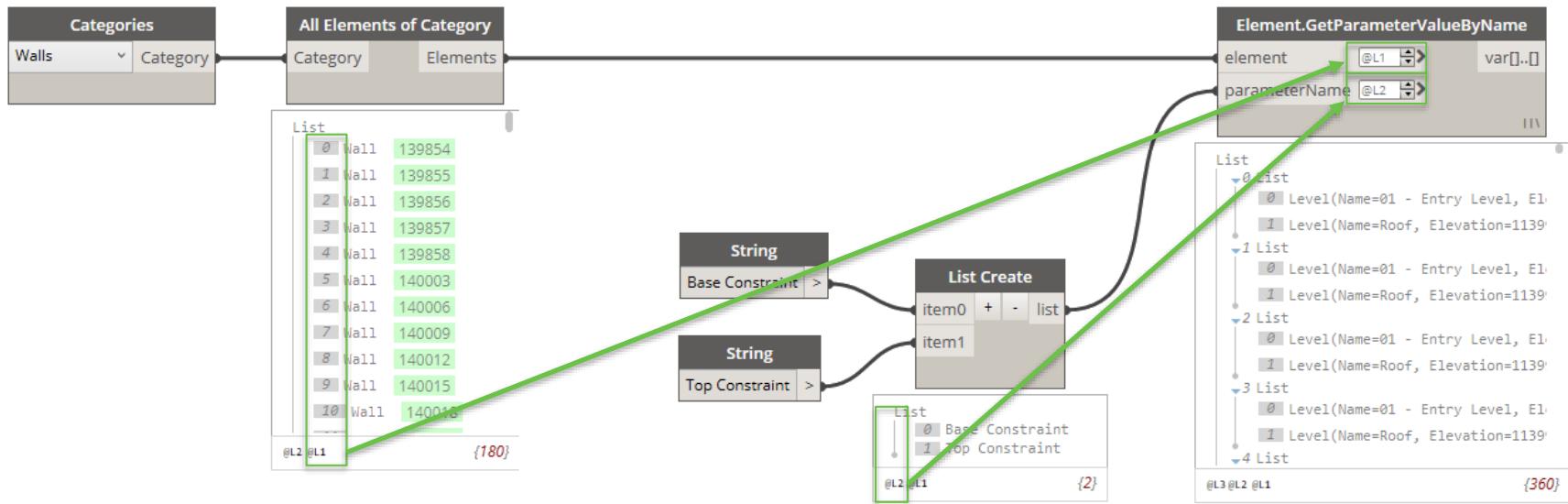
List@Level

First example



List@Level

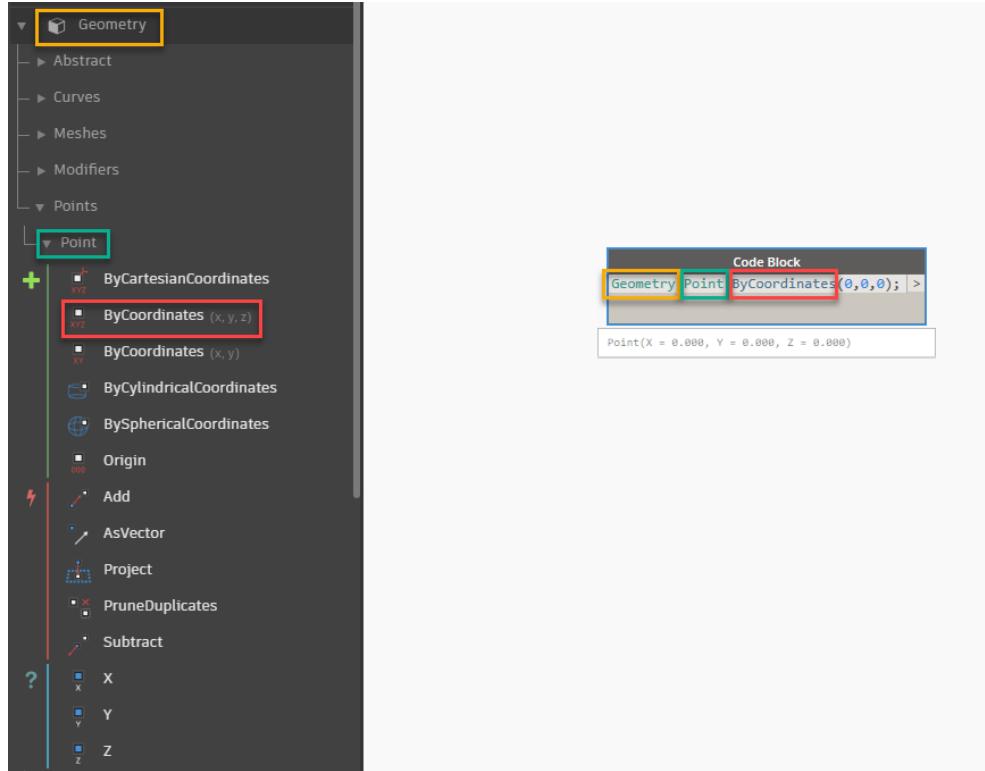
Second example



Code Blocks

- Dynamically links a visual programming environment with a text-based one.
- The code-block has access to all the Dynamo nodes
- Can define an entire graph or full parts of a graph in one node.
- Lets you manage all sorts of data and even create dictionaries

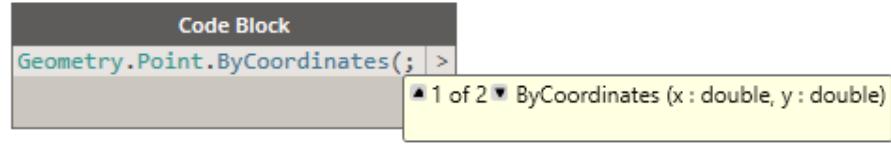
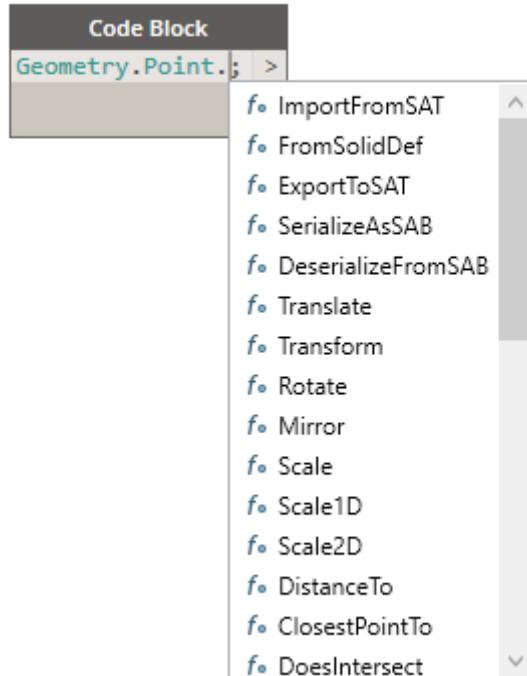
Code Blocks



Namespace : Geometry
Class : Point
Method/Property : ByCoordinates

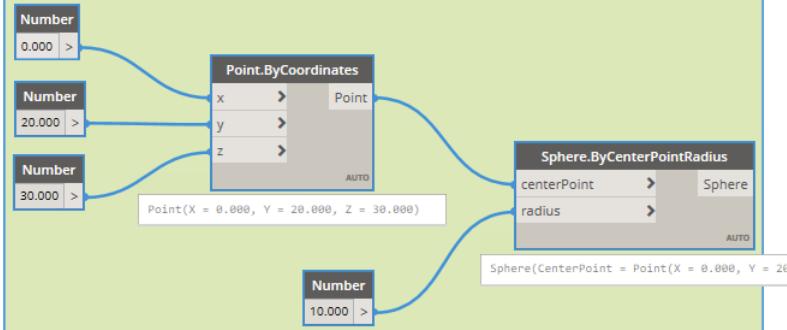
Code Blocks

Auto Completion

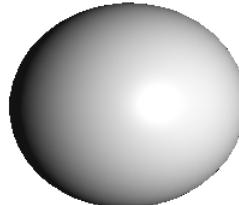
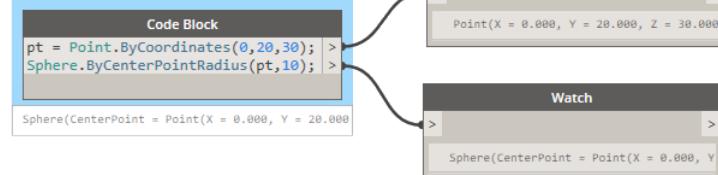


Code Blocks

Node version

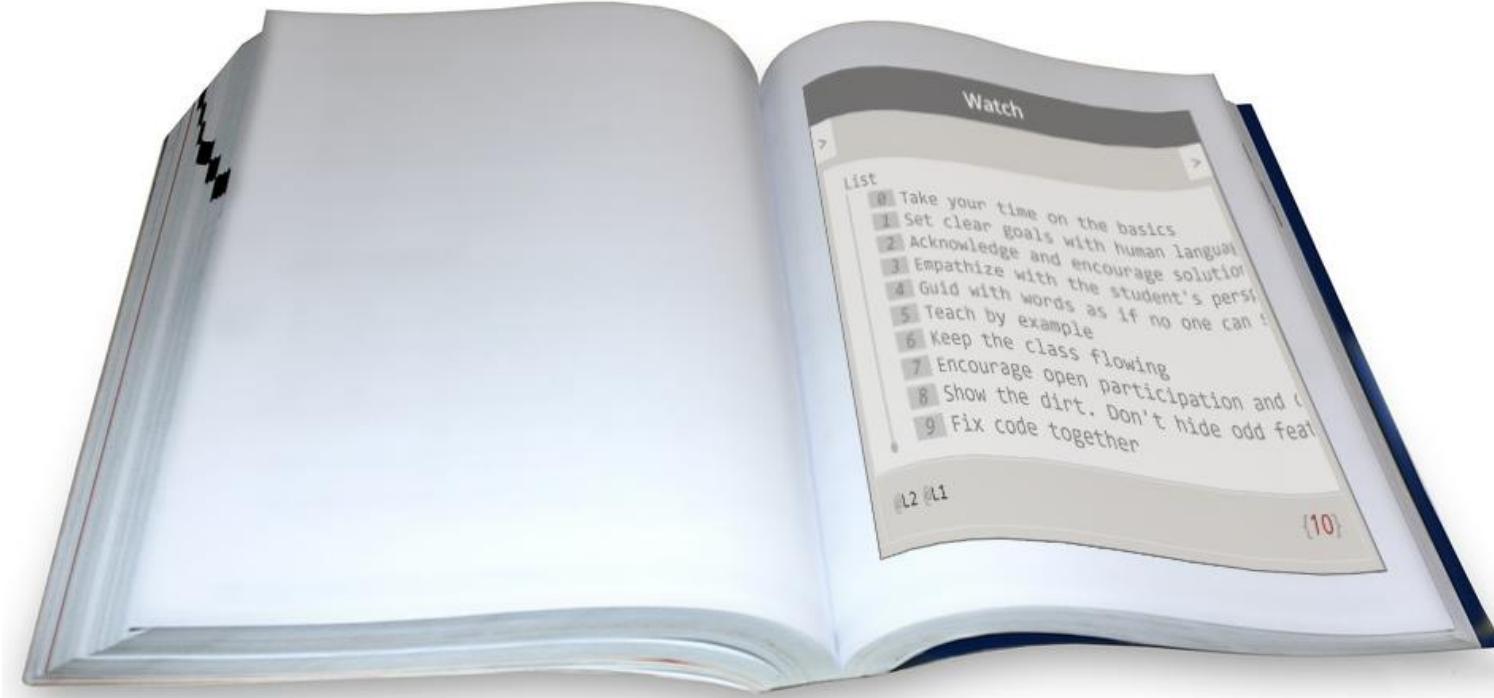


Code block - design script version



Best Practices On Teaching





<https://dynamobim.org/HowToTeachDynamo/>

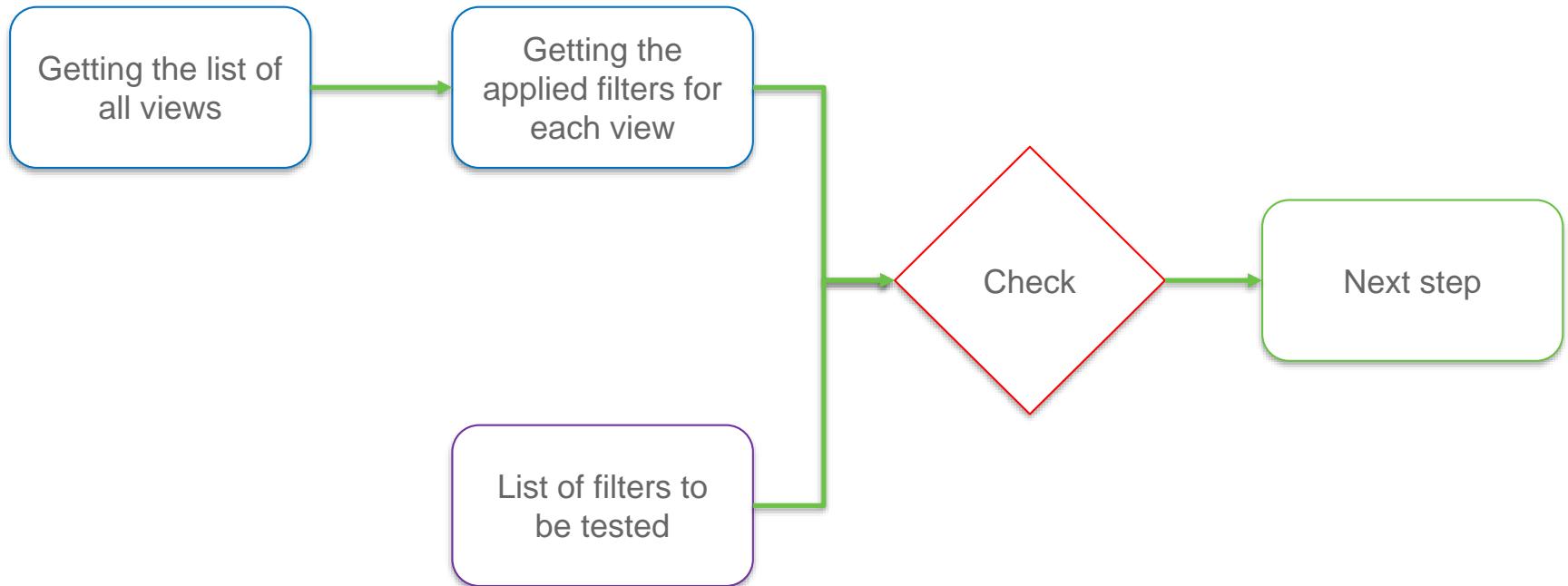
Some Advice



**First solve the
problem, then start
scripting.**

Draw a clear diagram of your strategy

Finding unused filters



**Encourage solution
diversity**

Let people know that there are often many solutions



**Using real life
analogies helps**

Cooking analogy

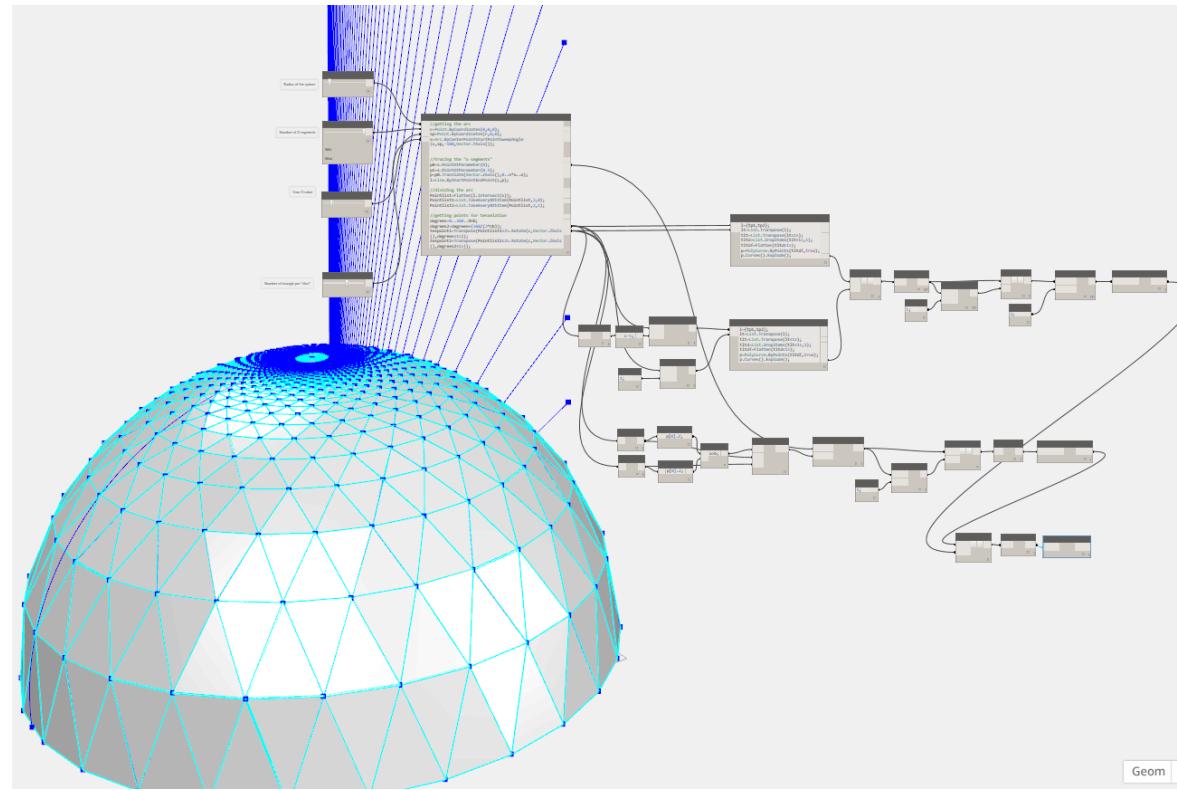


Construction analogy



**Don't show people
what YOU can do with
Dynamo.**

Too much at once can be intimidating



**Show them what
THEY can do with
Dynamo.**

Simple Real Life Use Cases



**Use real life use cases
that can help your
peers understand and
feel the power of
Dynamo**

For Architects

Renumbering Rooms

For Structural engineers

Placing piles on site topo

For MEP engineers

Generating Openings Automatically

For project managers

Getting Key Performance Indicators

File name

File size

Unplaced views

Warnings

For project managers

Getting Key Performance Indicators

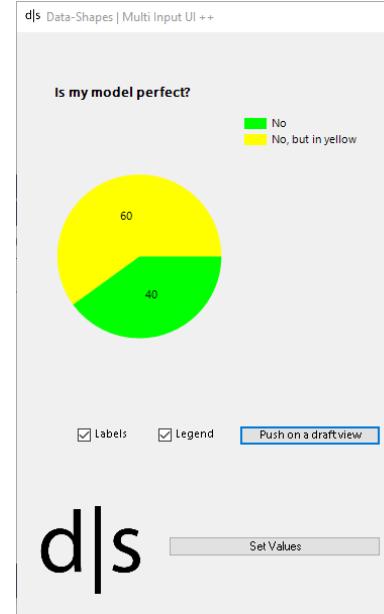
You Retweeted

jobN pfeison 🎉 60secondrevit

😎 Awesome charts from @DataShapes_io on #DynamoBIM for checking the quality of a #Revit model.



6:20 PM · Nov 14, 2019 · Twitter Web App



Nomenclature



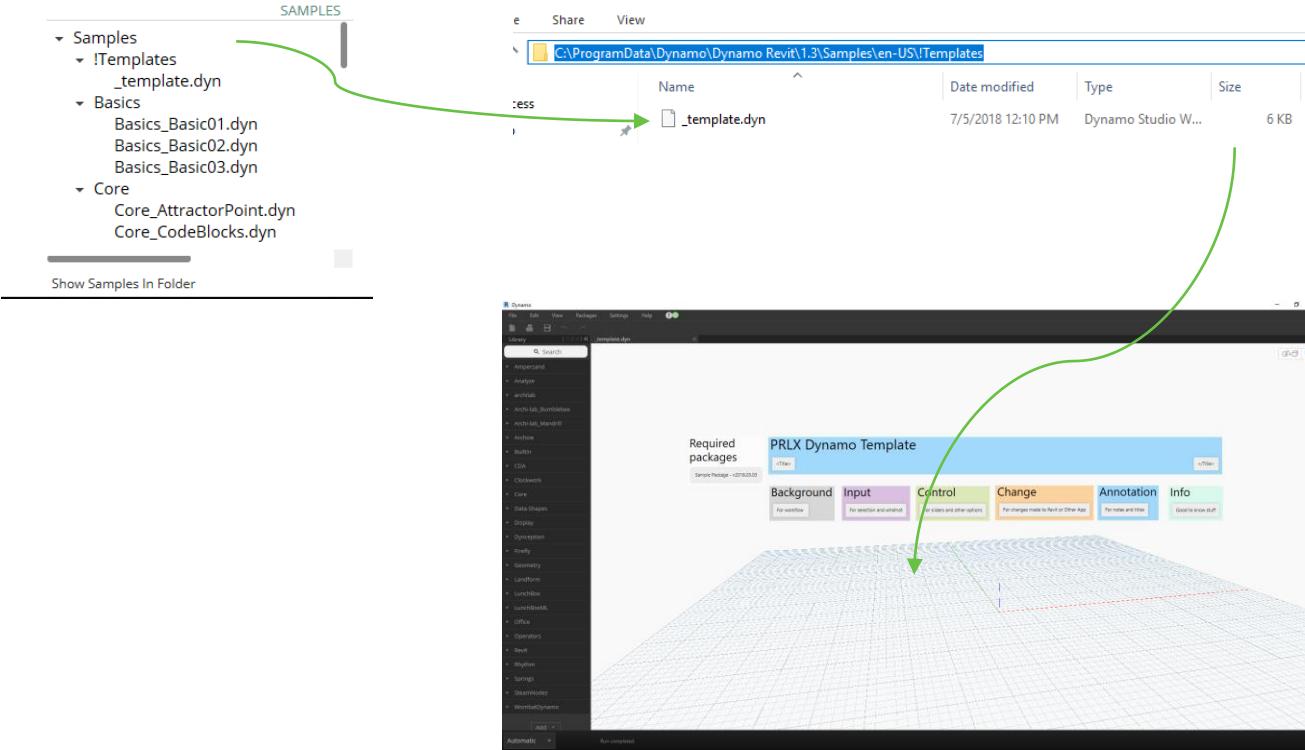
A wide-angle photograph of a large industrial warehouse. The perspective is down a long aisle between tall metal shelving units. The shelves are densely packed with various boxes, some labeled with red signs like "SIPPE" and "2". The ceiling is high with visible steel beams and overhead lighting. A red sign with the number "2" is visible on the left side of the aisle.

shippable scripts

Graph Building Strategies

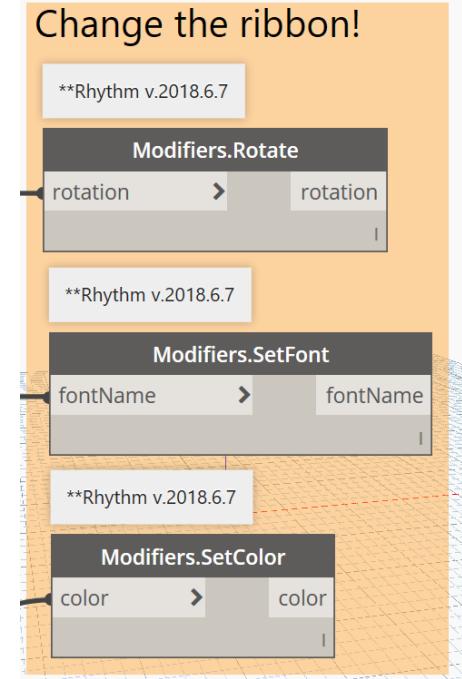


Strategy: Dynamo Template

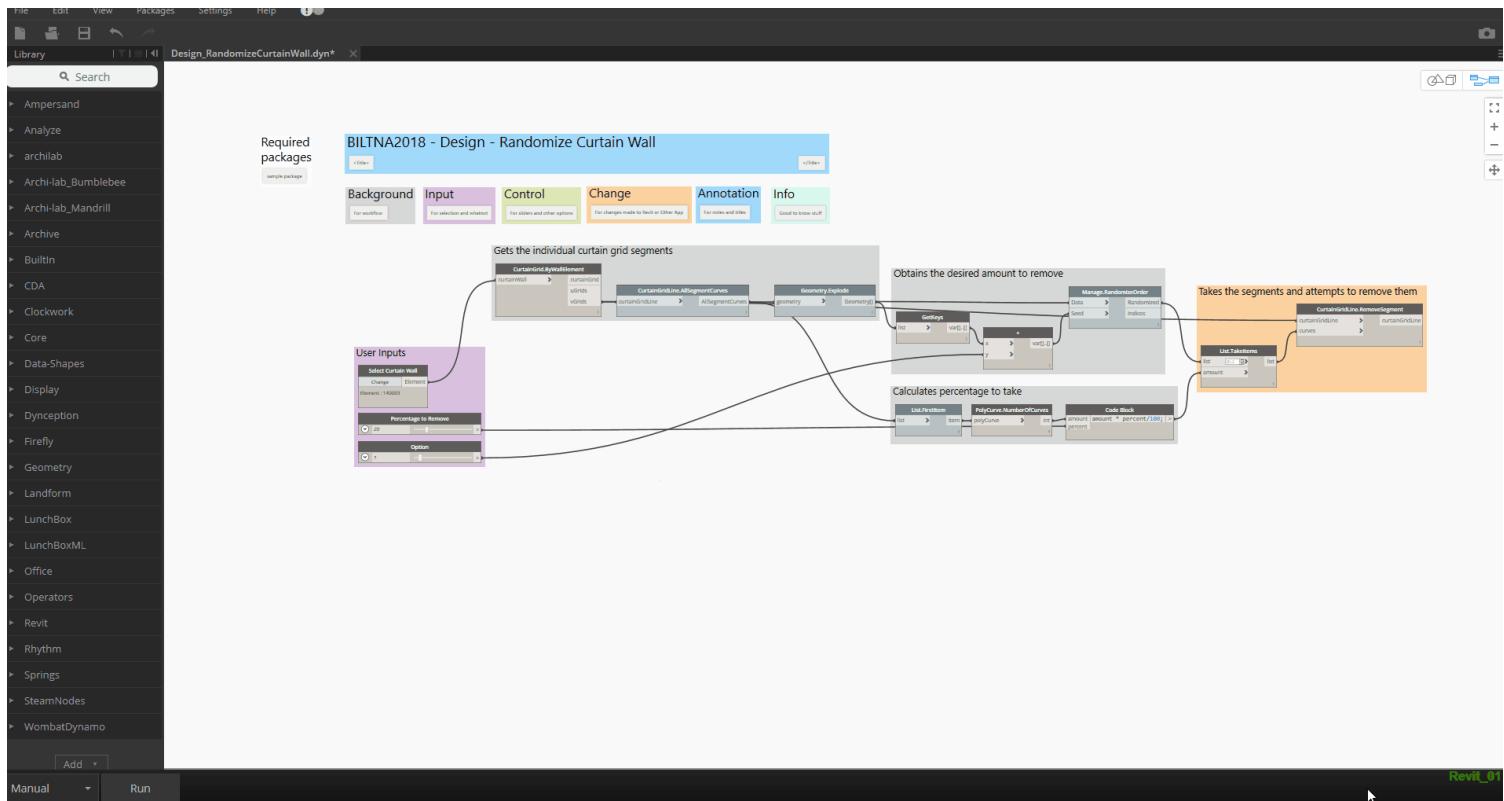


Strategy Annotating Nodes

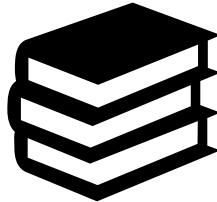
- Notes are added above each node (Ctrl+W on Keyboard or from View Tab).
- Standard is to start with “**” to indicate that it is a custom package note, (more on this in a bit).
- But, am I really expecting you to do this for EVERY node?

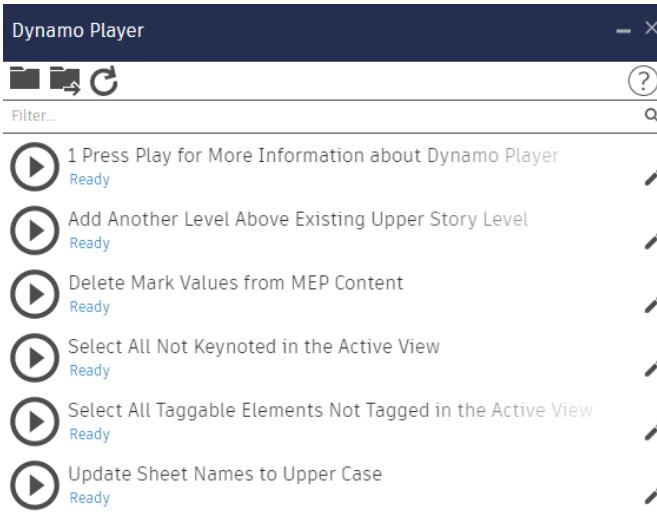


Yes I am. enter package usage doge



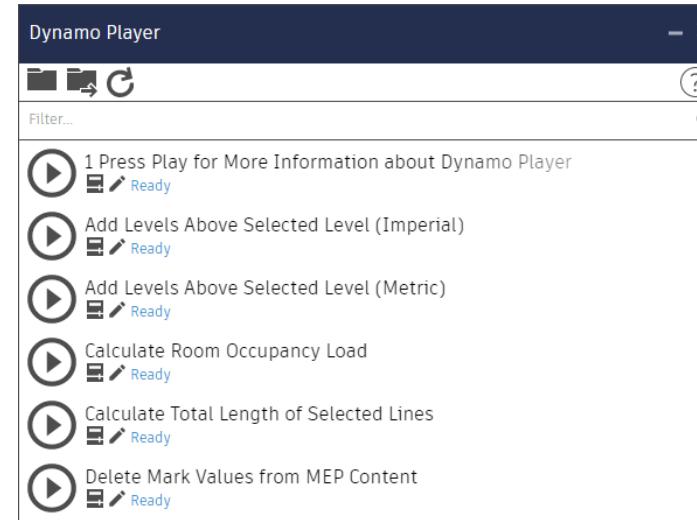
History of Dynamo Player





Revit 2017 Version

Revit 2017 introduced “Dynamo Player”, yay!
Now we can push buttons to run our graphs!
But without inputs..

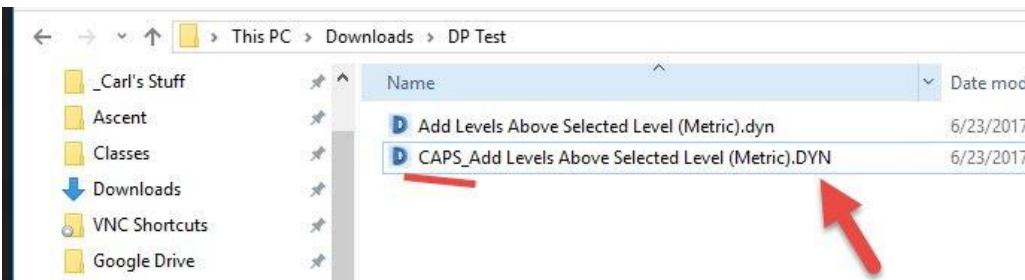


Revit 2018.1 Version

Revit 2018.1 updated Dynamo Player to allow inputs.
Ahhh, thank goodness.

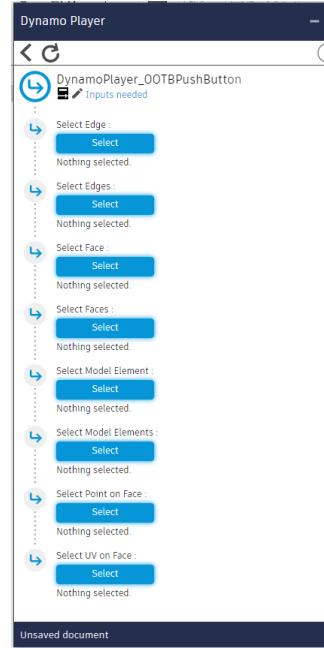
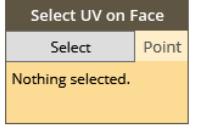
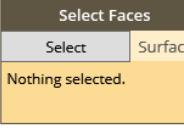
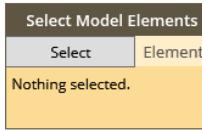
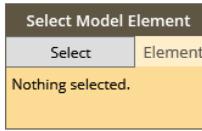
Considerations

File extensions with CAPITALS do not get recognized.

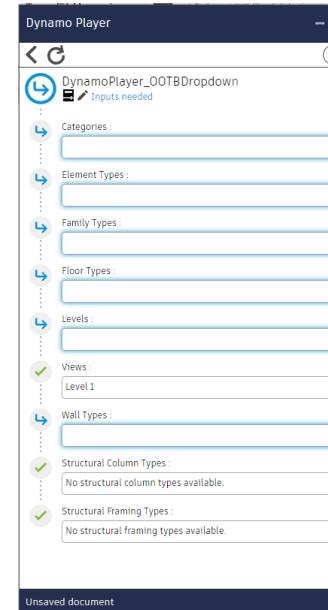
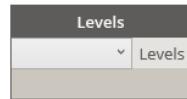
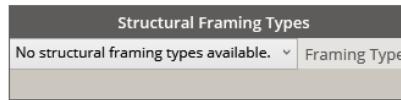
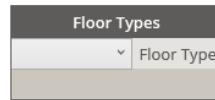
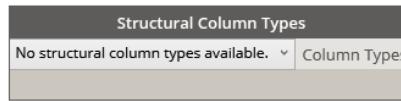
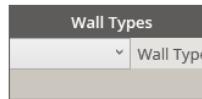
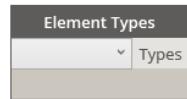
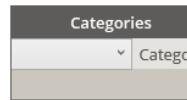


Credit: <https://twitter.com/theBIMsider/status/1011777948481937409>

Support for OOTB Push Button Nodes



Support for OOTB Drop Downs



Bonus Dropdowns

The screenshot displays the Revit ribbon interface with several dropdown menus open, illustrating the "Bonus Dropdowns" feature. The tabs shown are:

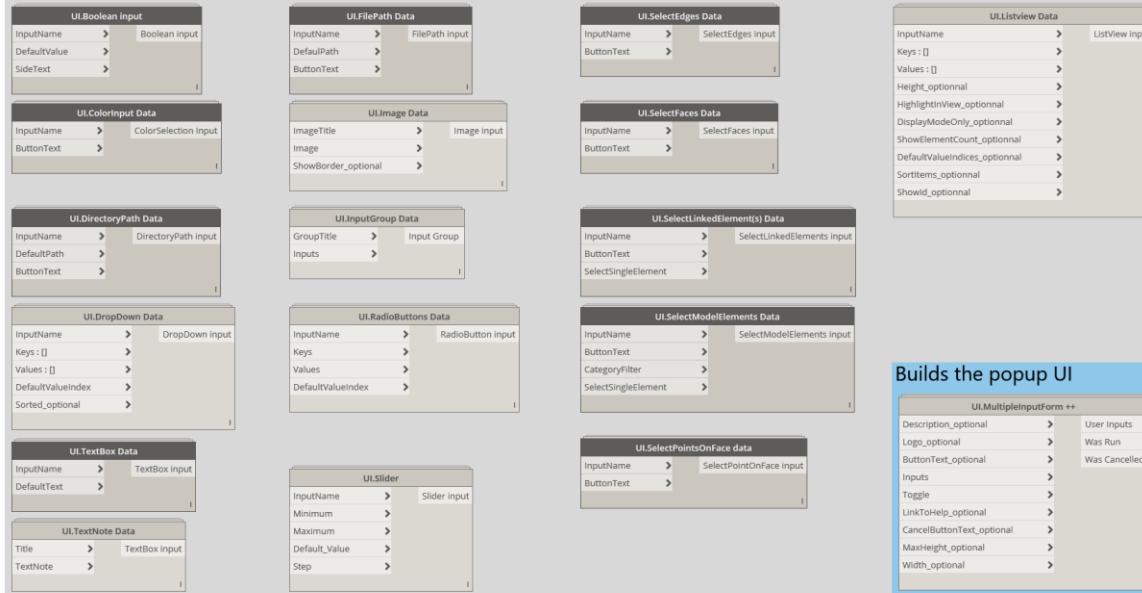
- Print + Export**: Contains dropdowns for Box Placement Types (BottomLeft), Image Resolution (DPI_150), Zoom Fit Type (FitToPage), Image File Type (BMP), and Export Range (CurrentView).
- Line Styles**: Shows a dropdown for lineStyle.
- View Type**: Shows a dropdown for viewType.
- Phase Filters**: Shows a dropdown for phaseFilter with an option to Show All.
- Duplicate Options**: Shows a dropdown for options with an option to AsDependent.
- Design Options**: Shows a dropdown for DesignOption with the message "No Design Options available in project.".
- All Elements of Model Category**: Shows a dropdown for element.
- Model Categories**: Shows a dropdown for Model Category.
- ViewFamilyTypes**: Shows a dropdown for ViewFamilyTypes.
- MEPSystemTypes**: Shows a dropdown for MEPSystemTypes with an option to CableTrayConduit.
- Links**: Shows a dropdown for Link with the message "No Links available in project.".
- Sheets**: Shows a dropdown for Sheet.
- Roof Types**: Shows a dropdown for Roof Type.
- SpaceTypes**: Shows a dropdown for SpaceType with an option to kActiveStorage.

archilab.net

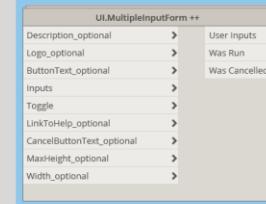
rhythm

Custom Dynamo Popup UI

Input Options

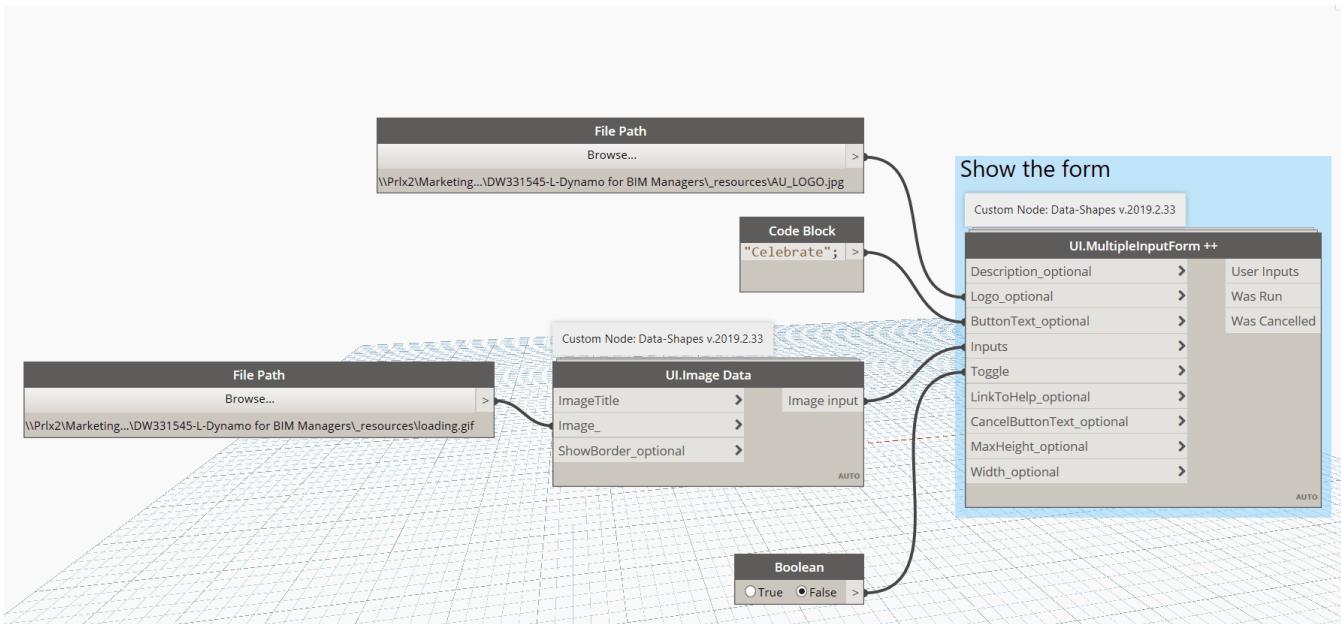


Builds the popup UI



d s

Custom UI allows us to....



Shippable Script 01 | 3D

Room Tags

- Good content = good workflow
- Think it out
- Fail-proofing your graph with Revit/Dynamo changes considered.

Shippable Script 02 | Remove Useless Room Separators

- Think out the logic involved (how to do it manually)
- Using a method of checking for difference between lists
- Build with OOTB Dynamo Player Feedback

Shippable Script 03 | Load View Templates from Resource

- Build out logic to work with background files
- Make wicked UI for our users
- Feedback via UI

A wide-angle photograph of a vast, flat landscape, likely a salt flat or coastal area, under a dramatic sky filled with orange and yellow clouds. A white van is driving from the bottom left towards the center of the frame, kicking up a cloud of dust. The foreground is a light-colored, textured surface.

a project's journey

**As a BIM Manager, it
is important to know
what can be done.**

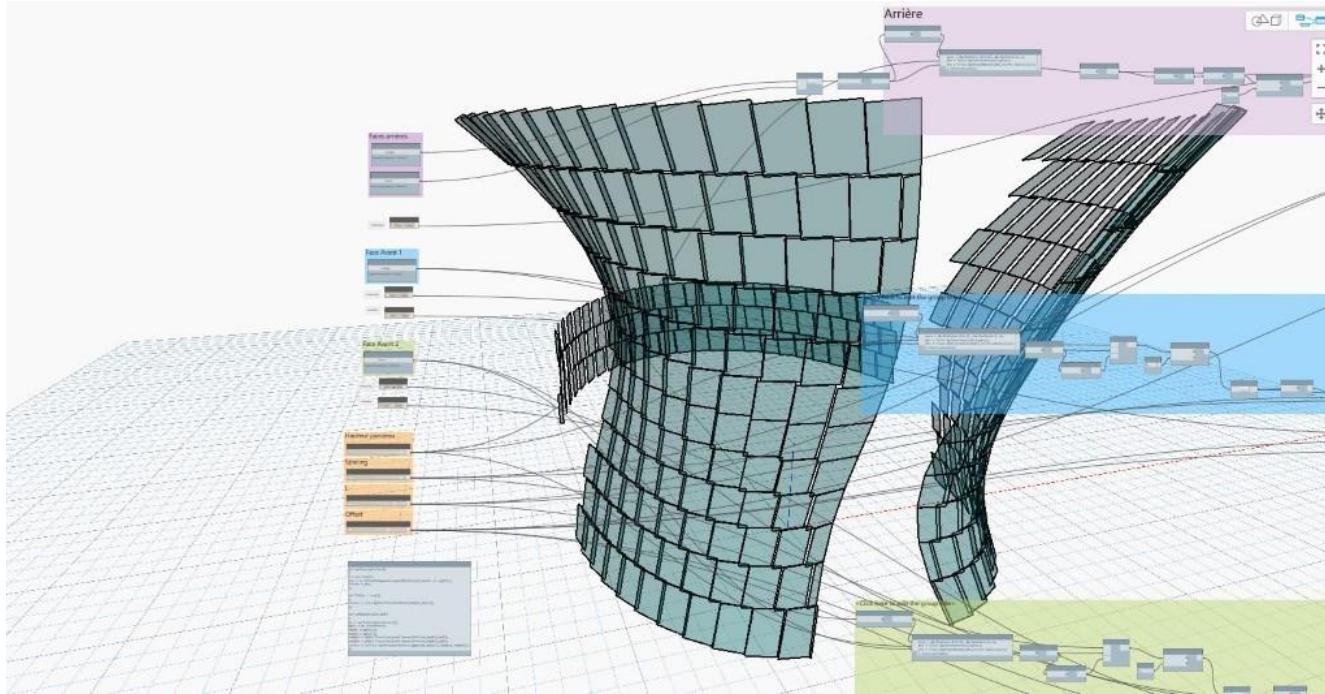
**You don't have to
know how to do
everything though...**

Early / Conceptual

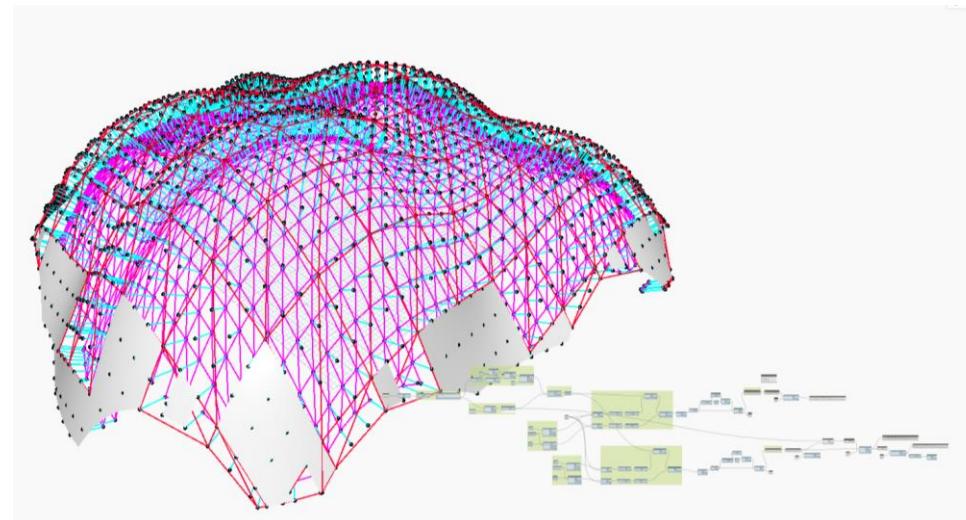
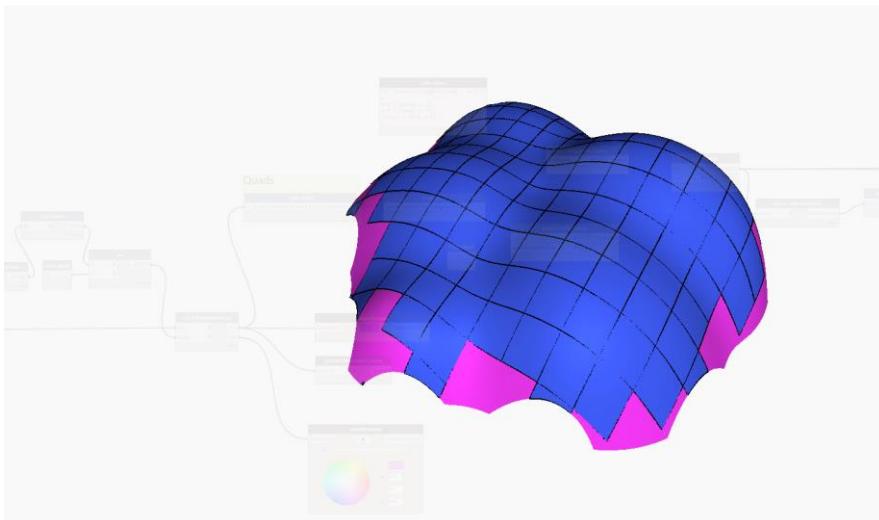


- + Awesomeness
- + Play with complexe geometries (tsplines)
- + interop with rhino/grasshopper
- + physics (dynashape)
- messiness of that type of script
- hard to standardize / generalize
- takes a special set of skills

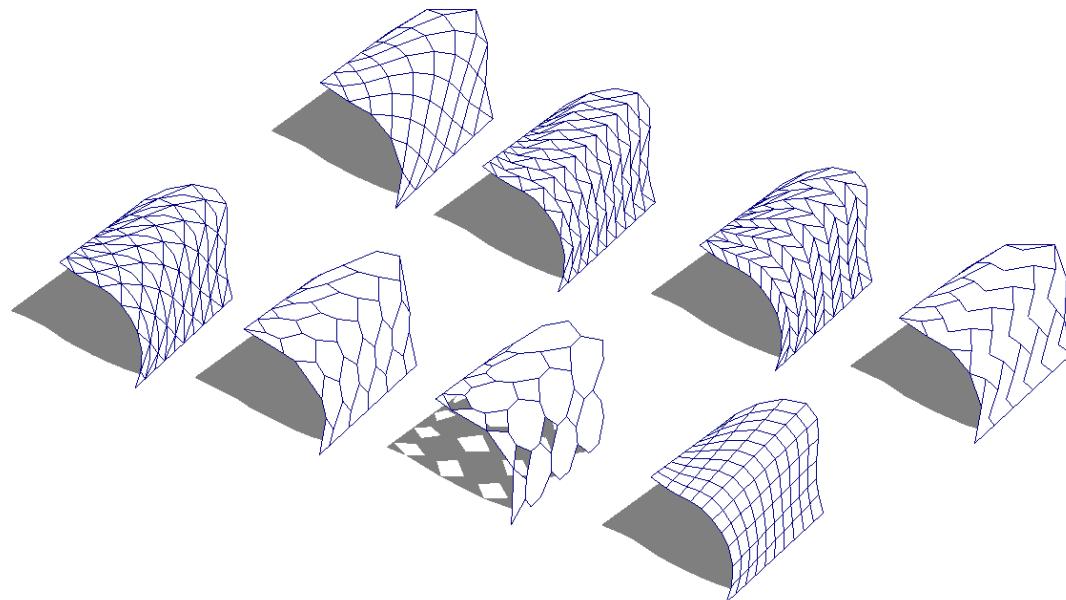
Early stage form finding / concept testing



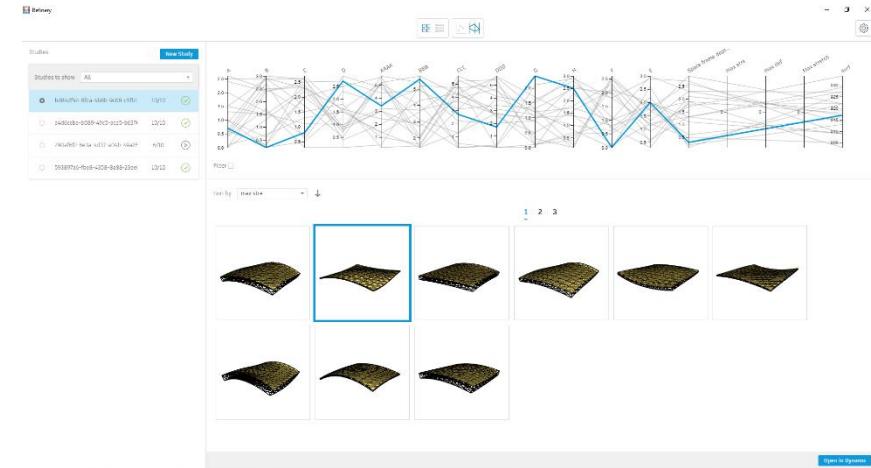
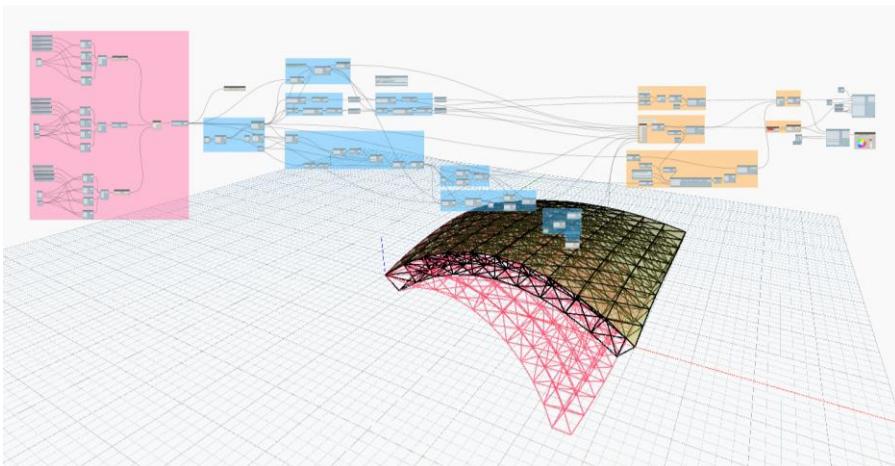
Early stage form finding / concept testing



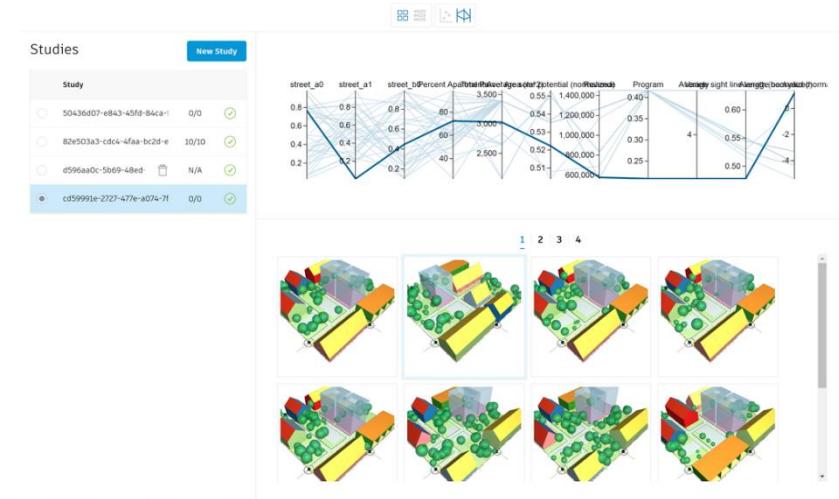
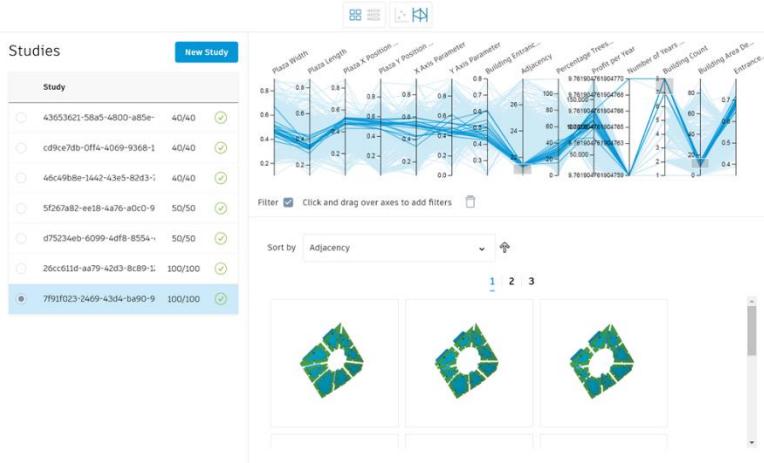
Early stage form finding / concept testing



Early stage form finding / concept testing



Early stage form finding / concept testing



Interoperability



AUTODESK
ROBOT



Production / Documentation

Automation



Automate renumbering (rooms, doors, sheets, cats , dogs)
Automate sheet layouts

KPIs

Model health checking , warning management

Deal with complexity

Mastering complex geometry, deal with complex
geometry for execution

Dynamo for advanced steel

QA / QC



Model health checking
Warning management
KPIs

Delivery





AUTODESK®

Make anything™

- Autodesk and the Autodesk logo are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries. All other brand names, product names, or trademarks belong to their respective holders. Autodesk reserves the right to alter product offerings, and specifications and pricing at any time without notice, and is not responsible for typographical or graphical errors that may appear in this document.
- © 2019 Autodesk. All rights reserved.

