

Intro Visual Programming with Revit/Dynamo

DIM_03

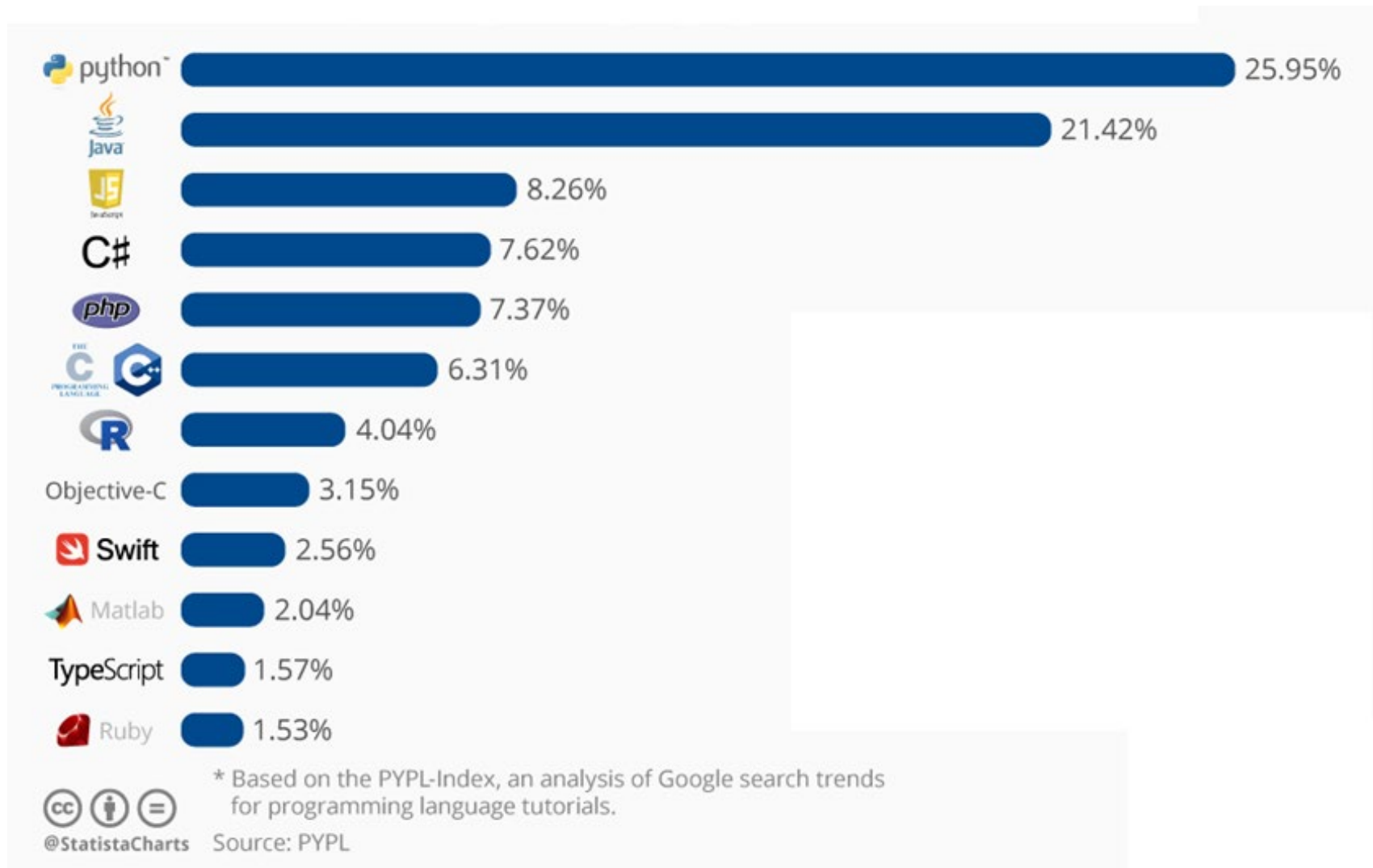
12.09.2024



Programming?



The most used languages?



Programming/Coding

Visual Scripting/Programming

Scripting

Why me?

That is not why I studied this or that.
I just want to use software.
I don't need to look under the hood.
The industry has far more difficult problems.
I don't need to understand coding.
Coding is so boring.

AI will solve the coding in future -
I just need to use programs.

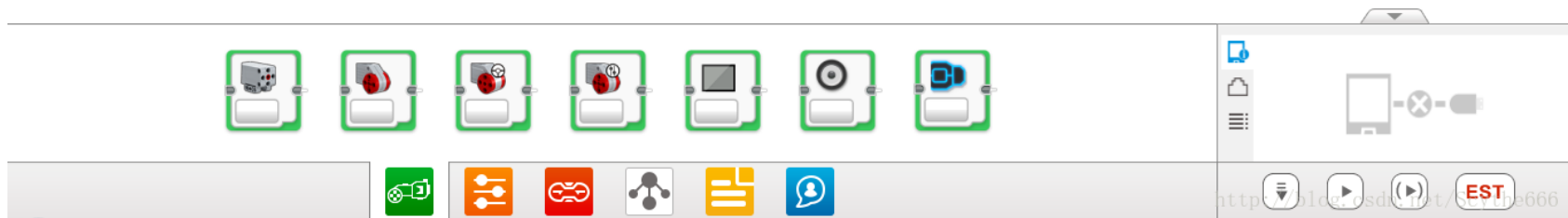
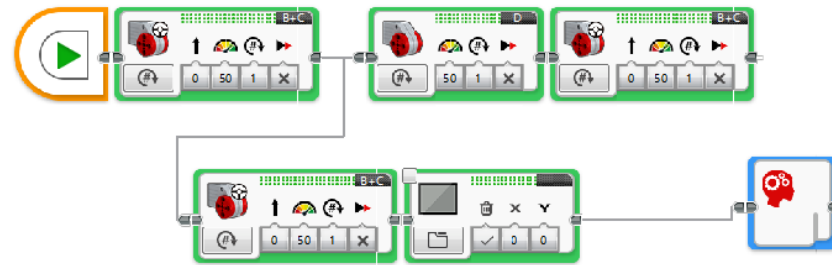
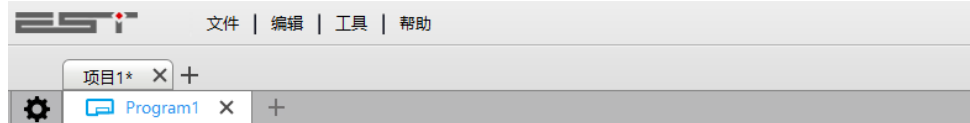
Coding is killing creativity
and (mostly) not suitable for architects

**Be aware of
your future colleagues!**

scratch



lego/mindstorms

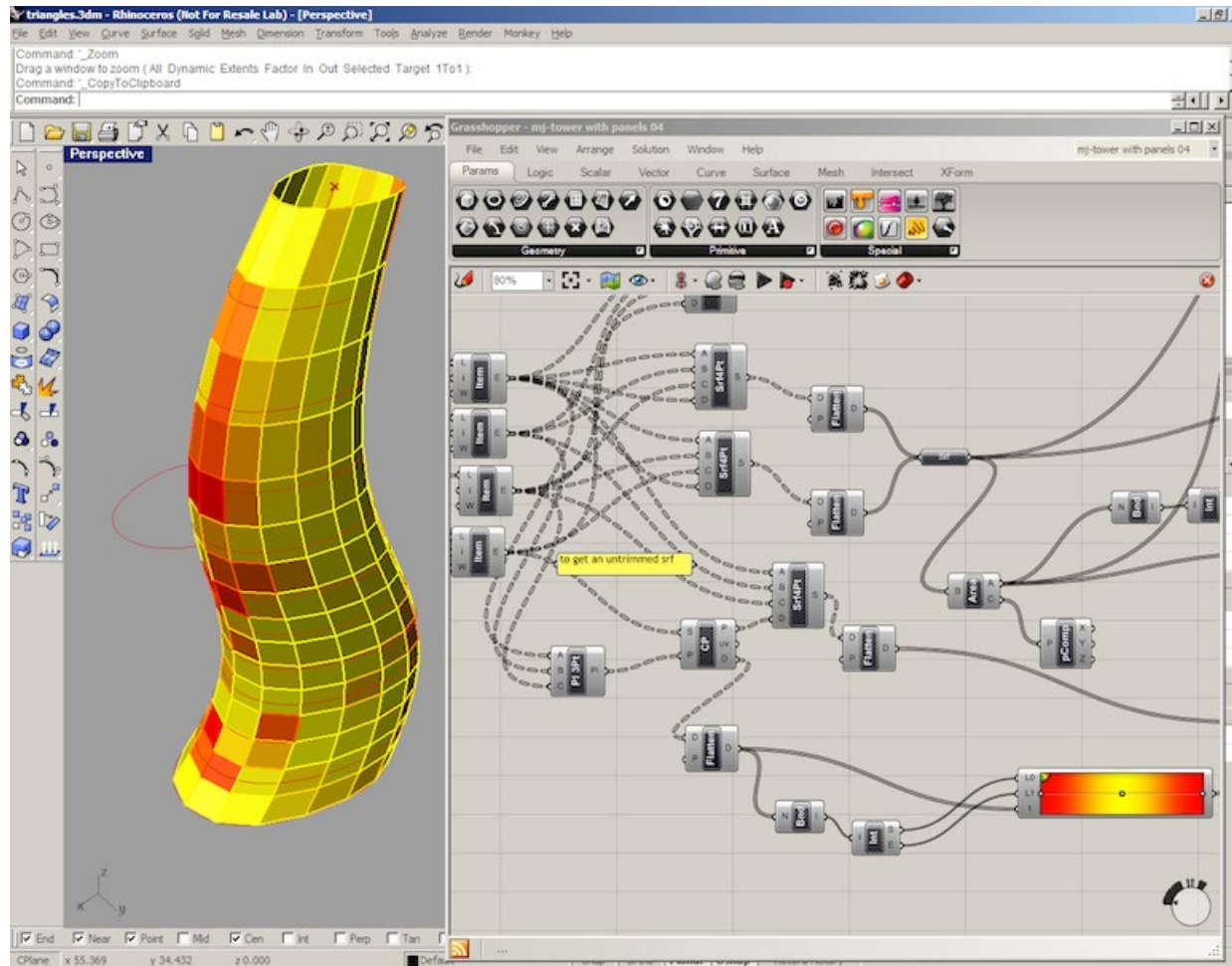


google/blockly

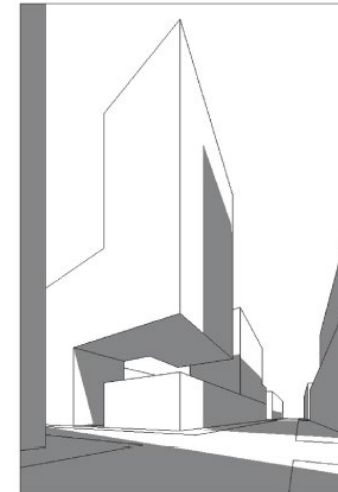
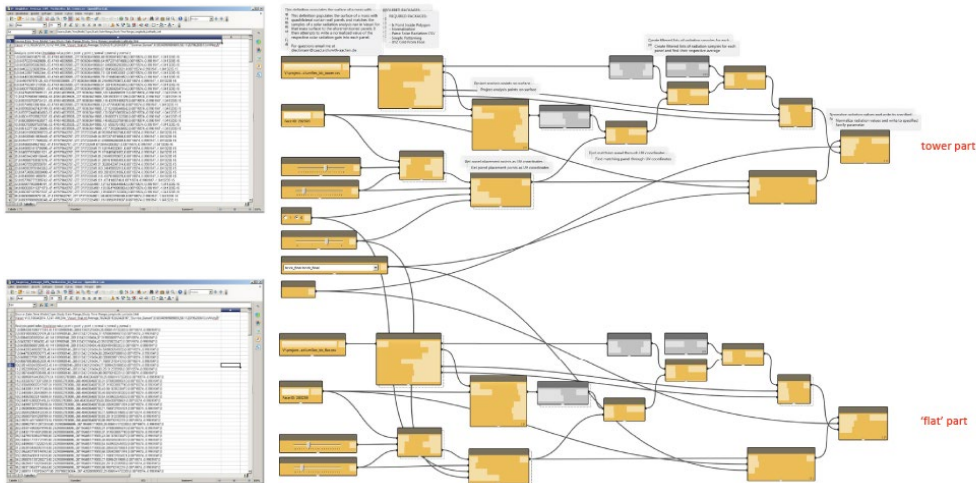
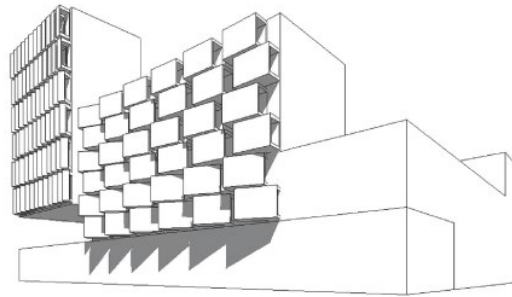
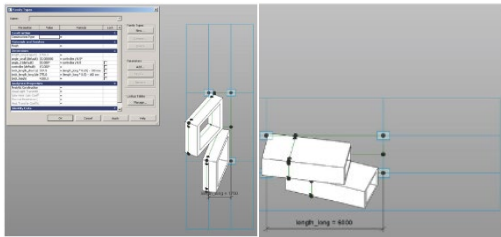
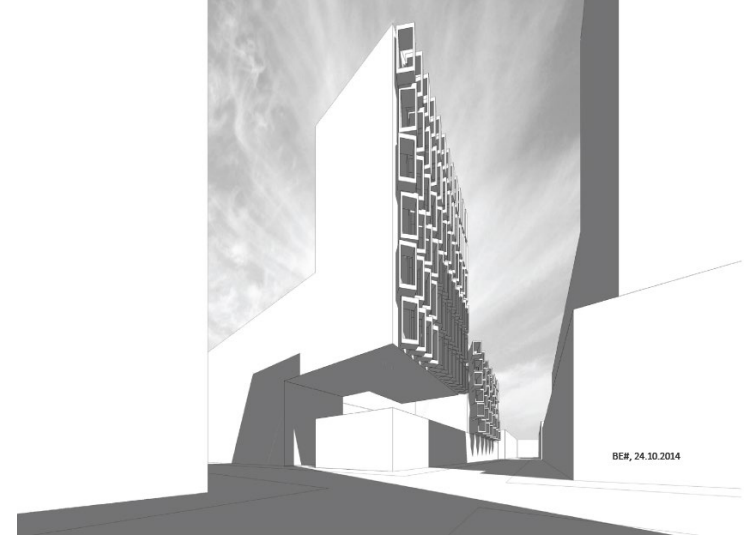
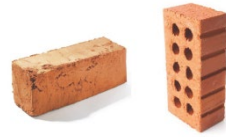
The screenshot shows the 'Blockly Games: Maze' interface. At the top, the browser address bar displays the URL <https://blockly-games.appspot.com/maze?lang=en&level=10&skin=0>. The page title is 'Blockly Games : Maze' with a progress indicator of 10 out of 10 levels. The language is set to 'English'. The main area is divided into three sections:

- Maze:** A maze with a yellow path. A green robot is at the start, and a red pin marks the goal. A message at the bottom left says 'You have 4 blocks left.' and a red 'Reset' button is below it.
- Code Editor:** A list of blocks on the left and a script on the right. The script is a 'repeat until' loop with a red pin icon, containing:
 - do if path to the left (with a dropdown arrow)
 - do turn left (with a dropdown arrow)
 - do if path ahead (with a dropdown arrow)
 - do move forward
 - else turn right (with a dropdown arrow)

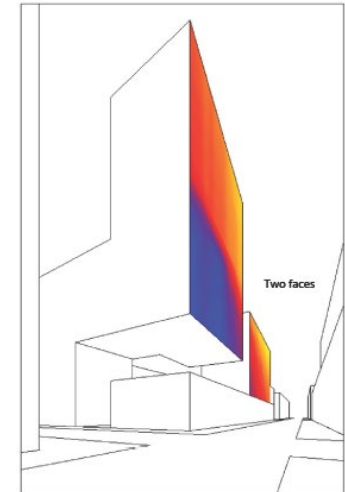
rhino/grasshopper



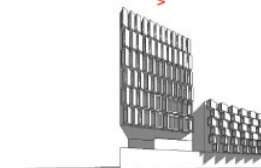
revit/dynamo



vertical



Two faces



horizontal

Dynamo?



what is dynamo?

Google

dynamo org



Search

All

News

Images

Videos

Maps

More

Settings

Tools

About 40.200.000 results (0,43 seconds)

Dynamo BIM

<https://dynamobim.org>

Computational Design. **Dynamo** extends building information modeling with the data and logic environment of a graphical algorithm editor. GET **DYNAMO**→.

Download

Sandbox is a free download of our core technology that isn't ...

Learn

Dynamo BIM. Learn ... Learn how to open and interact with ...

The Dynamo Primer

Download the Dynamo v1.3 Primer here.

[More results from dynamobim.org »](#)

Blog

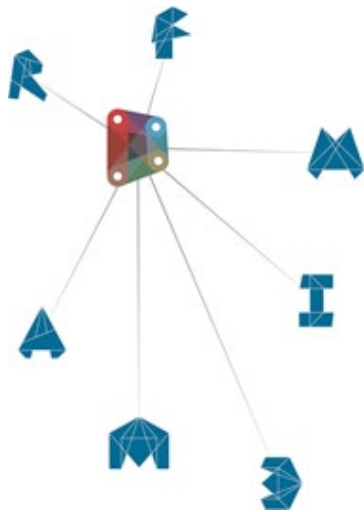
The DynaSpace package is developed by Long Nguyen ...

Explore

Dynamo is an open-source software platform for ...

What is Dynamo?

Dynamo is, quite literally, what you make it. Working with Dynamo ...



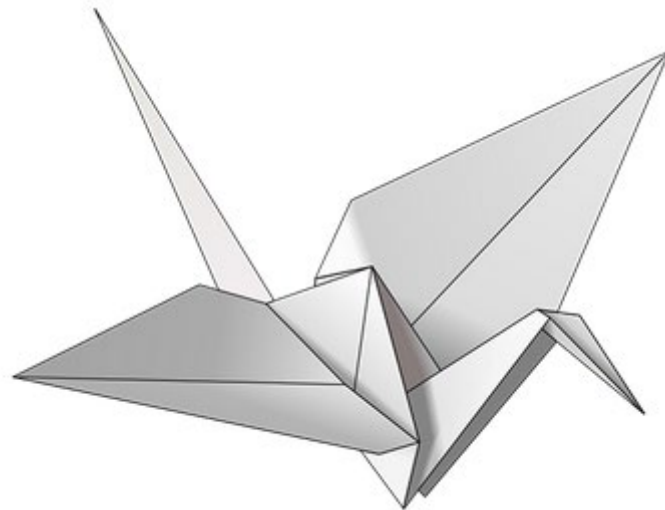
https://dynamobim.org/

The image shows a screenshot of the dynamobim.org website. The top navigation bar includes links for Learn, Explore, Blog, Forum, Developer, and Resources, along with social media icons and a 'GET DYNAMO' button. The main banner features the text 'programming for design' over a background of architectural models.

Below the banner, there are two main sections:

- Computational Design:** A section with a circular logo of colored dots and text stating 'Dynamo extends building information modeling with the data and logic environment of a graphical algorithm'.
- Download Section:** Two columns of download options:
 - DYNAMO SANDBOX:** Described as an open source environment for visual programming. It includes a 'DOWNLOAD' button and the version '2.17.0'. A list of features includes: testing up-to-date features, standalone operation, package management, multi-threaded performance, and dependency requirements.
 - DYNAMO REVIT:** Described as a graphical programming interface for Revit. It includes a list of features: rapid design iteration, lightweight scripting, and availability for Revit 2017-2019. A note states it is automatically installed as part of Revit since 2020. A red box highlights a link to 'Dynamo Builds' in the text: 'you can find the Dynamo4Revit installers on [Dynamo Builds](#)'.

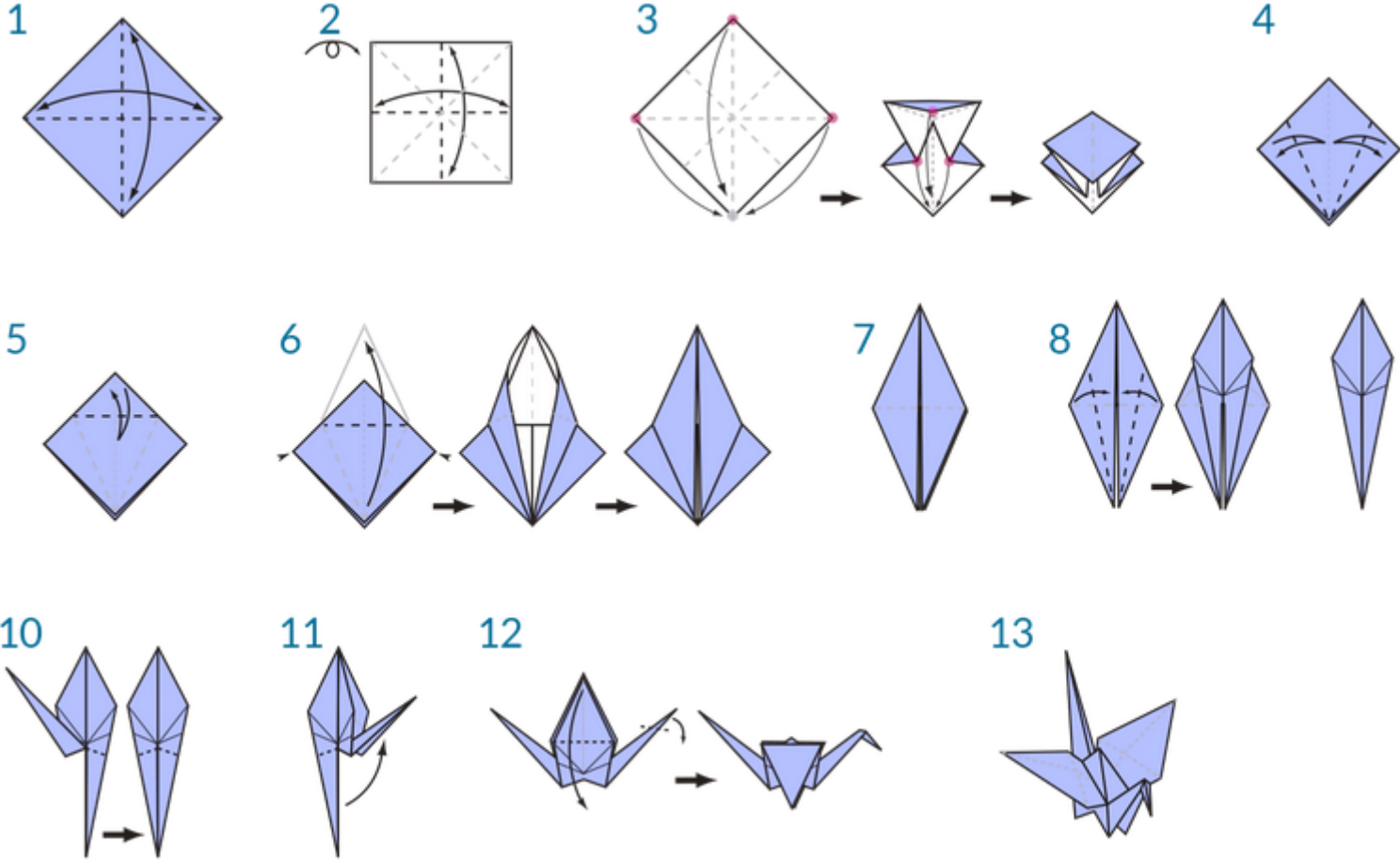
let's assume



textual instructions

- Start with a square piece of paper, colored side up. Fold in half and open.
Then fold in half the other way.
- Turn the paper over to the white side. Fold the paper in half, crease well and open,
and then fold again in the other direction.
- Using the creases you have made, Bring the top 3 corners of the model down to the bottom corner.
Flatten model.
- Fold top triangular flaps into the center and unfold.
- Fold top of model downwards, crease well and unfold.
- Open the uppermost flap of the model,
bringing it upwards and pressing the sides of the model inwards at the same time. Flatten down, creasing well.
- Turn model over and repeat Steps 4-6 on the other side.
- Fold top flaps into the center.
- Repeat on other side.
- Fold both 'legs' of model up, crease very well, then unfold.
- Inside Reverse Fold the "legs" along the creases you just made.
- Inside Reverse Fold one side to make a head, then fold down the wings.
- You now have a crane.

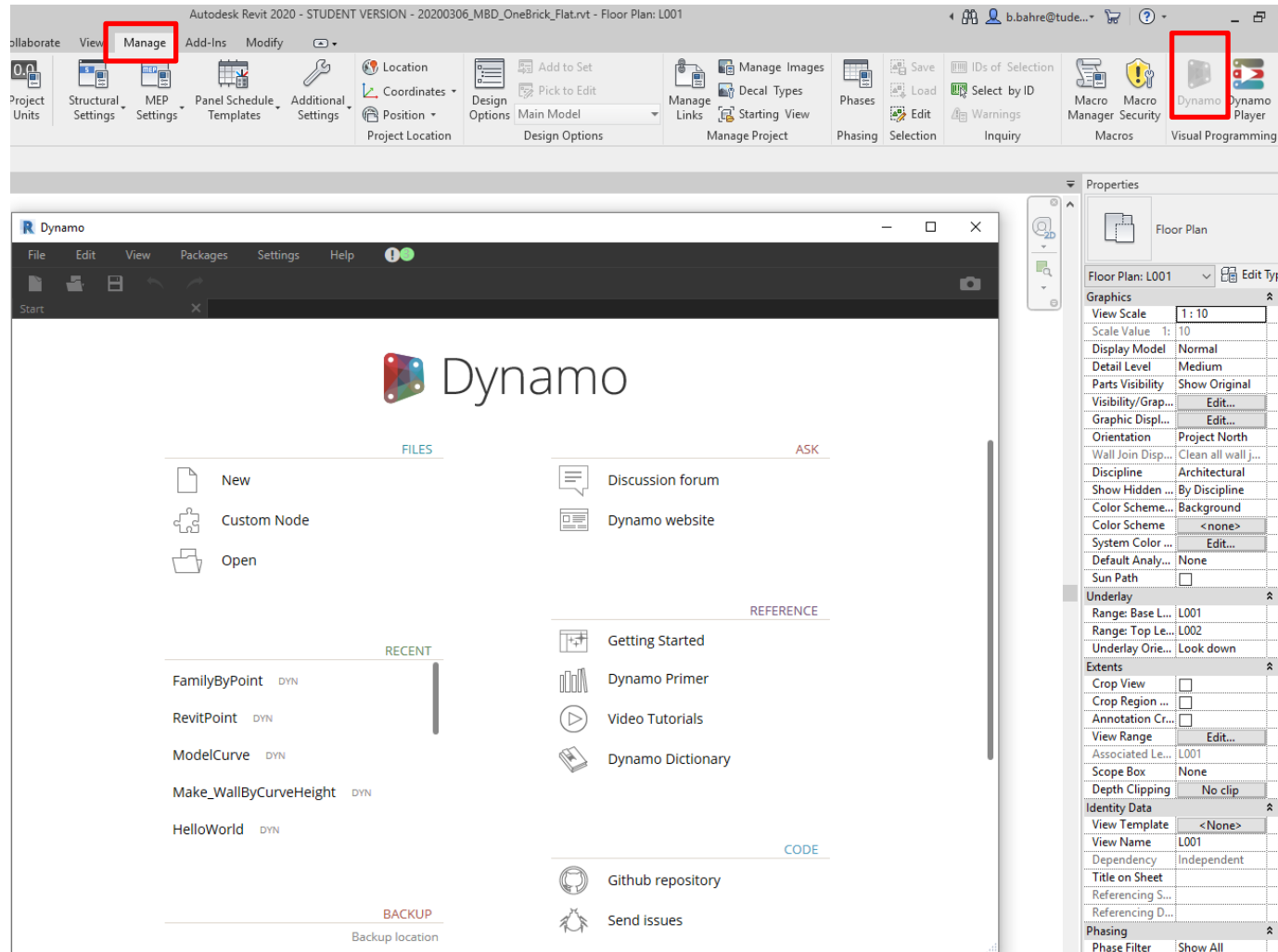
graphical instructions



Caution

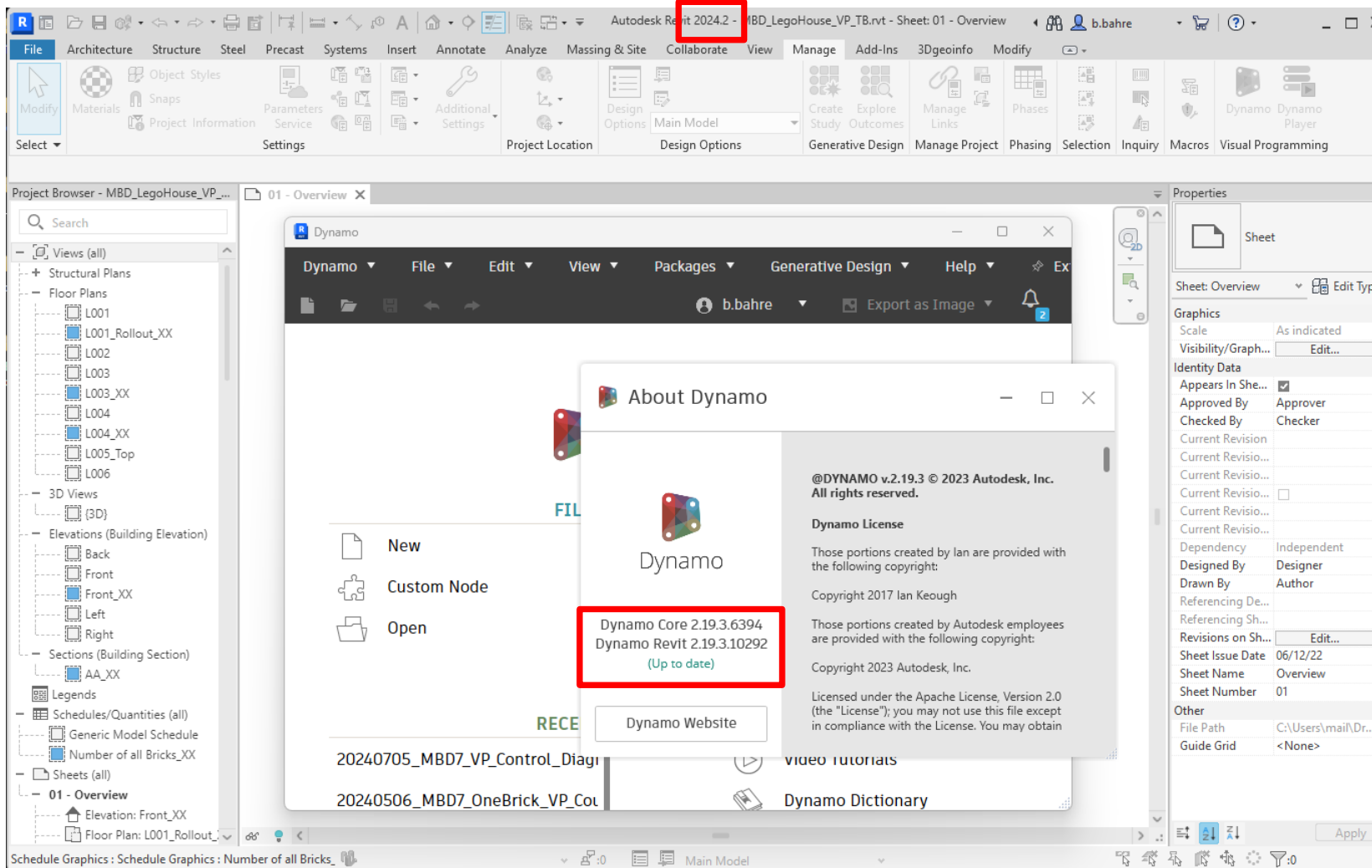


How to start

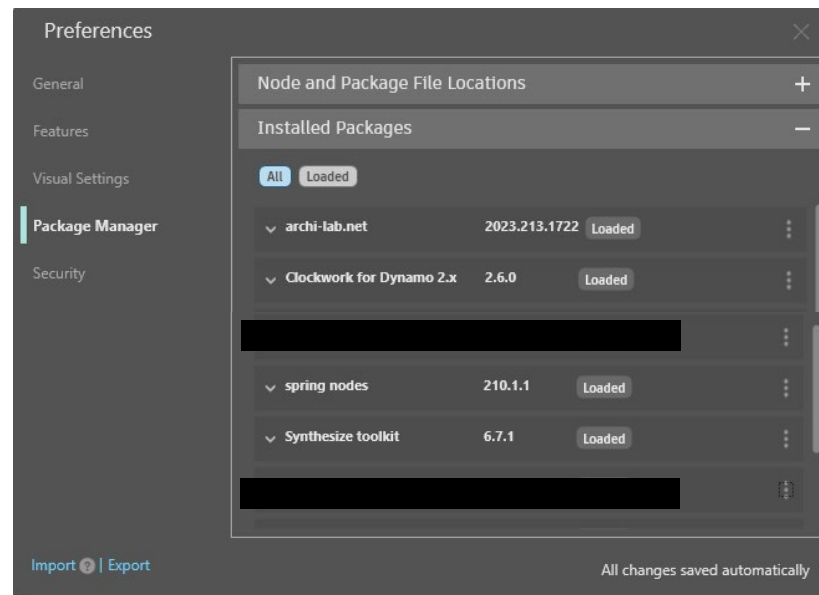


Check the version

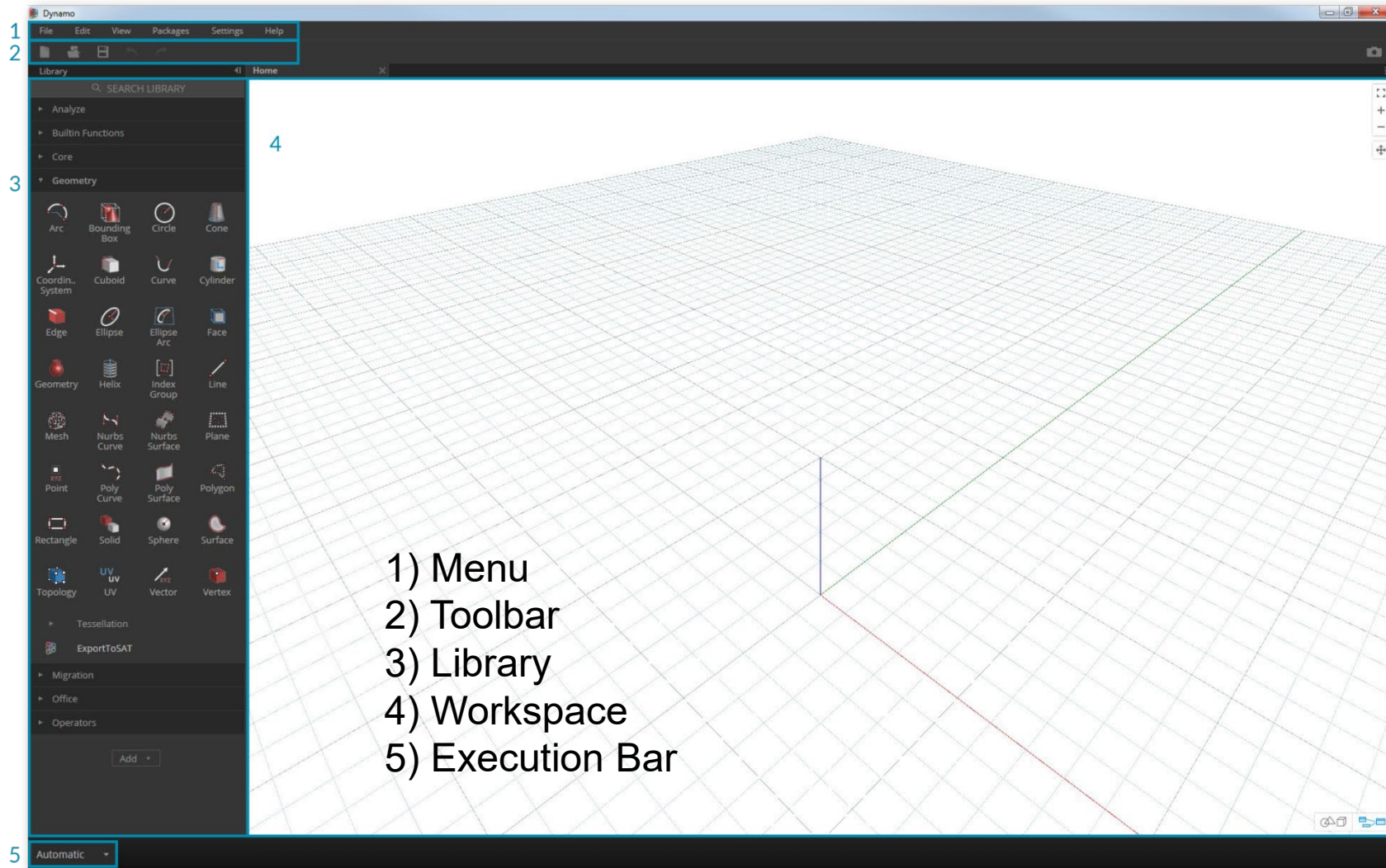
Check that version is 'Up to Date'



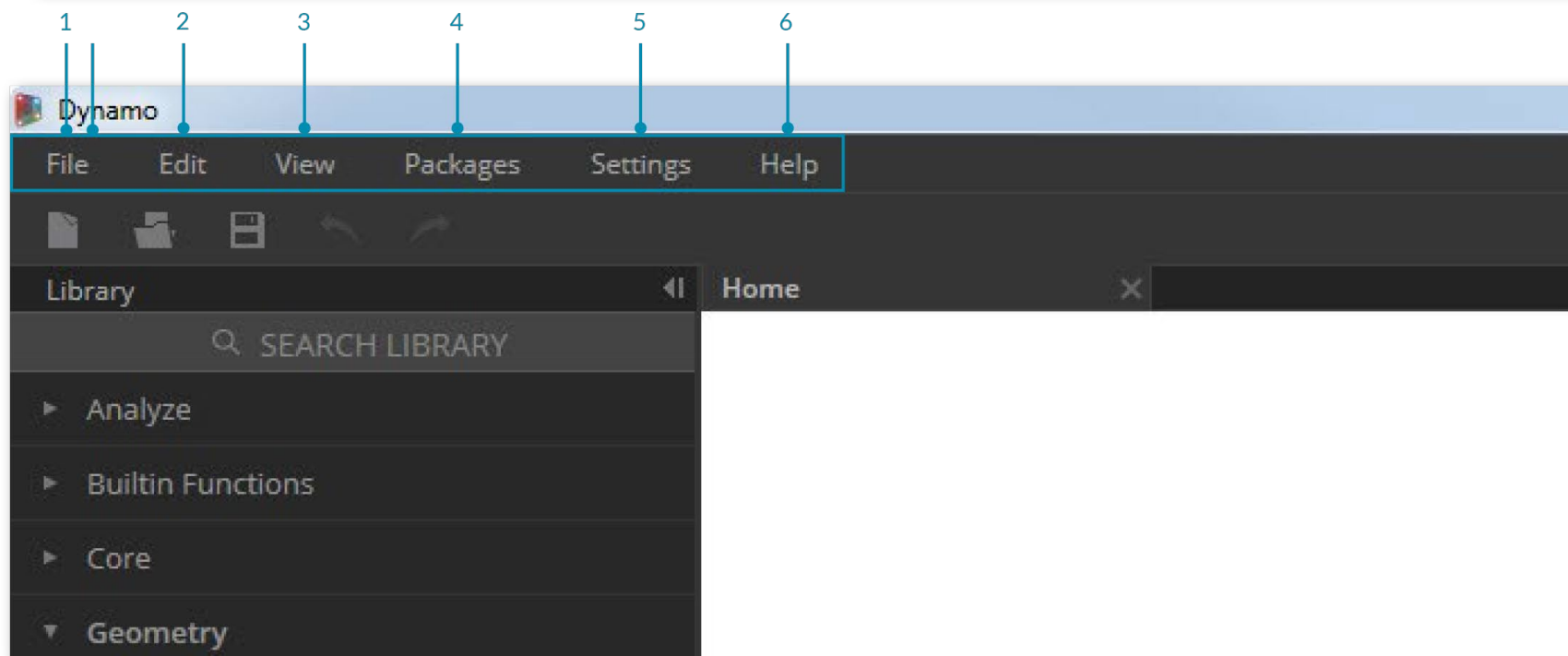
Needed Add-ons



the UI

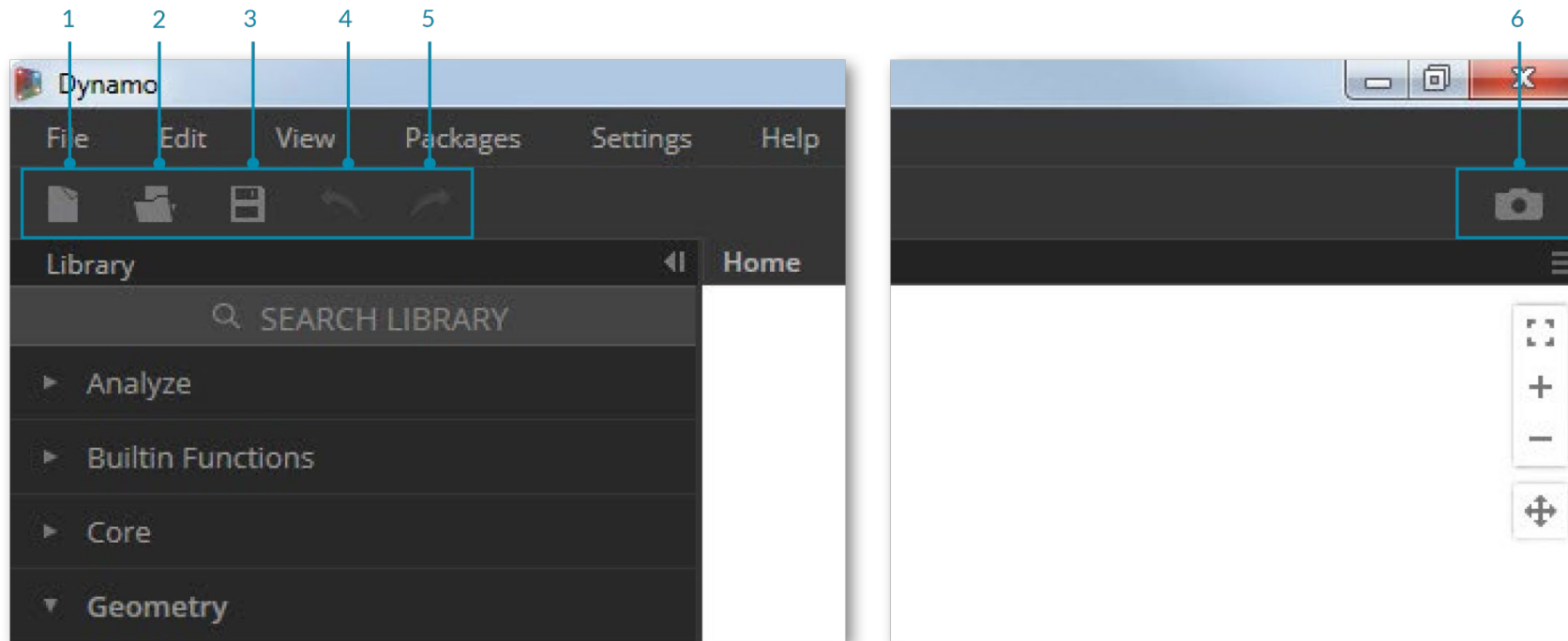


menu's



- 1) File
- 2) Edit
- 3) View
- 4) Packages
- 5) Settings
- 6) Help

toolbar



- 1) New - Create a new .dyn file
- 2) Open - Open an existing .dyn (workspace) or .dyf (custom node) file
- 3) Save/Save As - Save your active .dyn or .dyf file
- 4) Undo - Undo your last action
- 5) Redo - Redo your the next action
- 6) Export Workspace as Image - Export the visible workspace as a PNG file

library

The image shows a screenshot of a 3D software library interface. On the left, there is a dark sidebar with a search bar and a list of categories. The main area is a 3D grid with a blue vertical line, a green diagonal line, and a red diagonal line. A list of categories is overlaid on the grid.

1) Analyze
3) Builtin Functions
3) Core
4) Geometry
5) Migration
6) Office
7) Operators

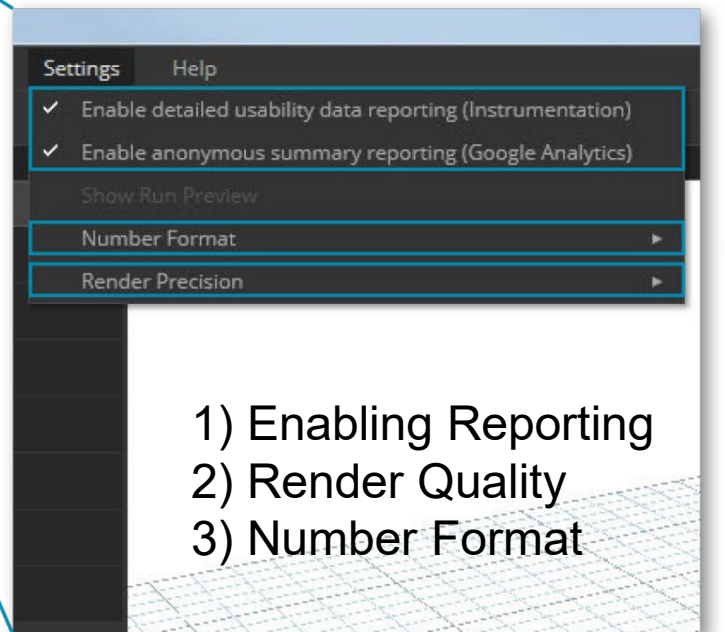
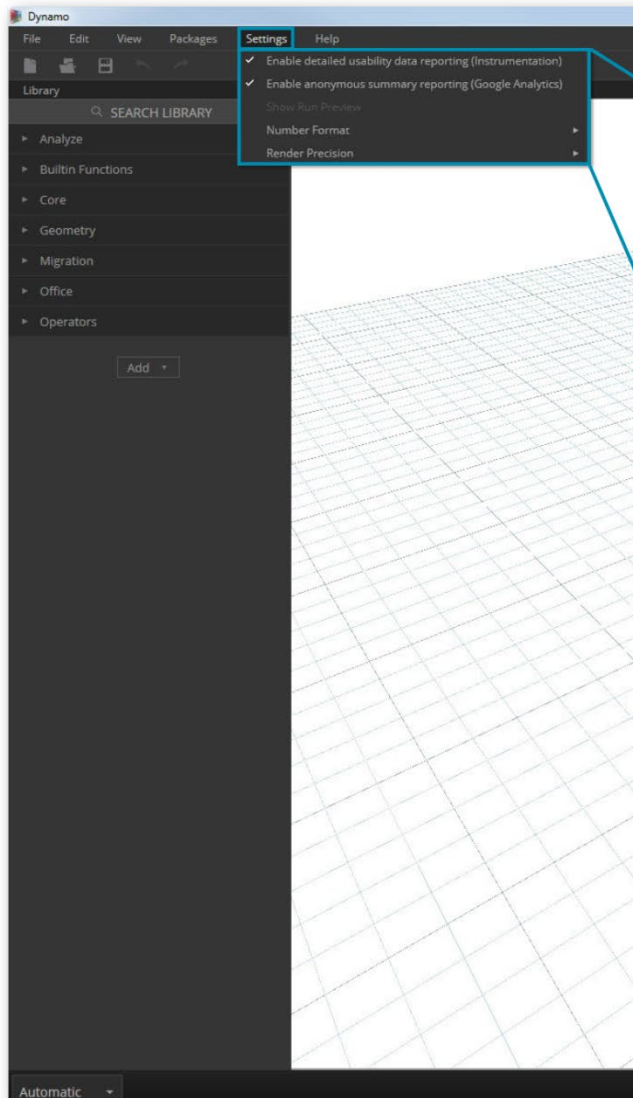
1
2
3
4
5
6
7

Library SEARCH LIBRARY Home

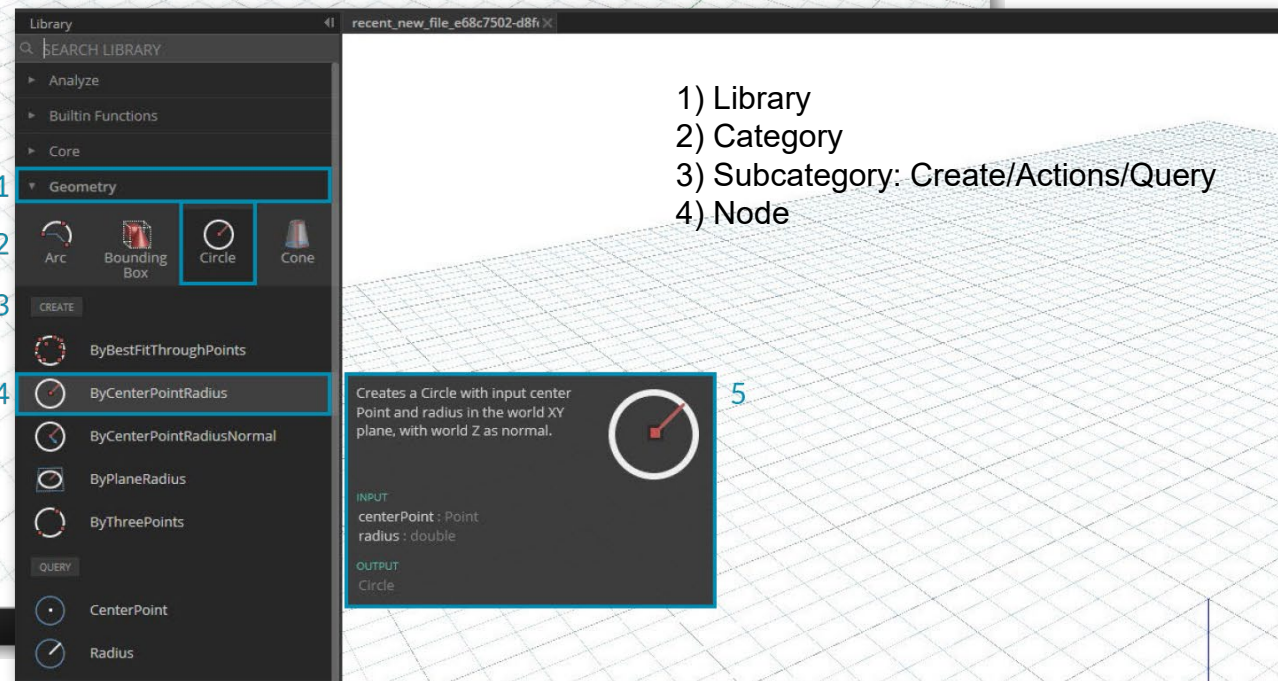
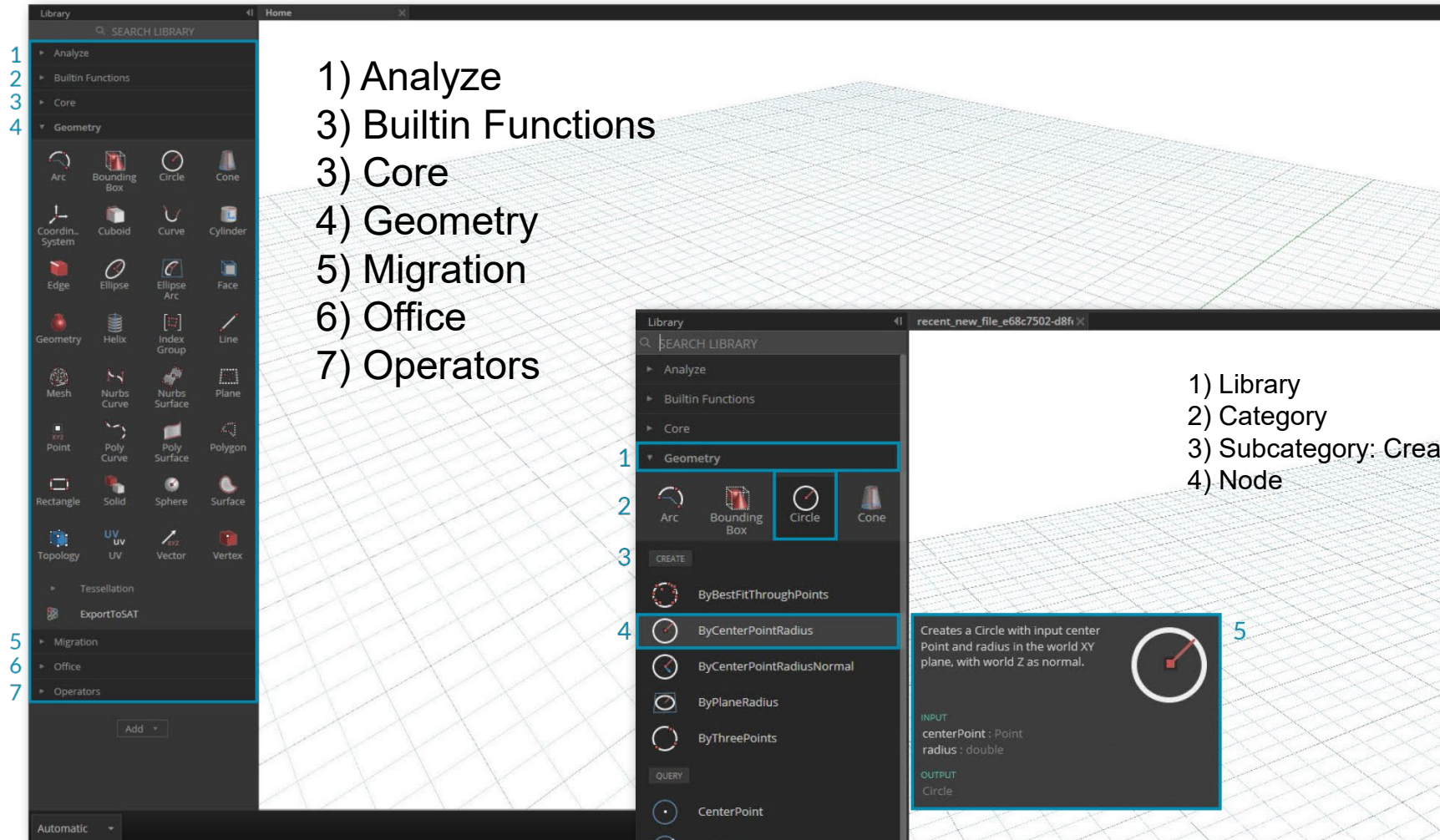
Analyze
Builtin Functions
Core
Geometry
Arc Bounding Box Circle Cone
Coordin.. System Cuboid Curve Cylinder
Edge Ellipse Ellipse Arc Face
Geometry Helix Index Group Line
Mesh Nurbs Curve Nurbs Surface Plane
Point Poly Curve Poly Surface Polygon
Rectangle Solid Sphere Surface
Topology UV UV Vector Vertex
Tessellation
ExportToSAT
Migration
Office
Operators
Add

Automatic

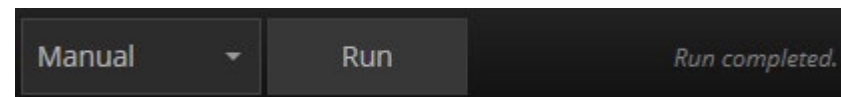
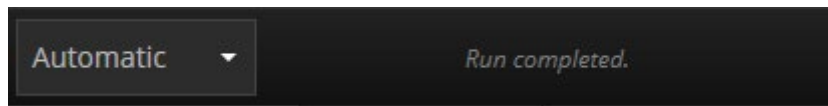
settings



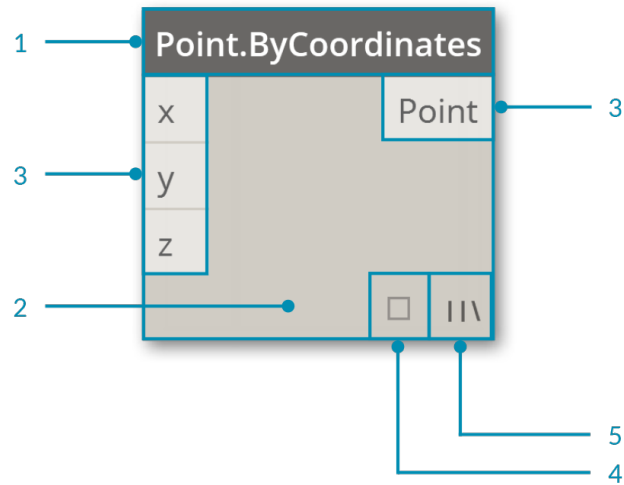
library



Run manual!



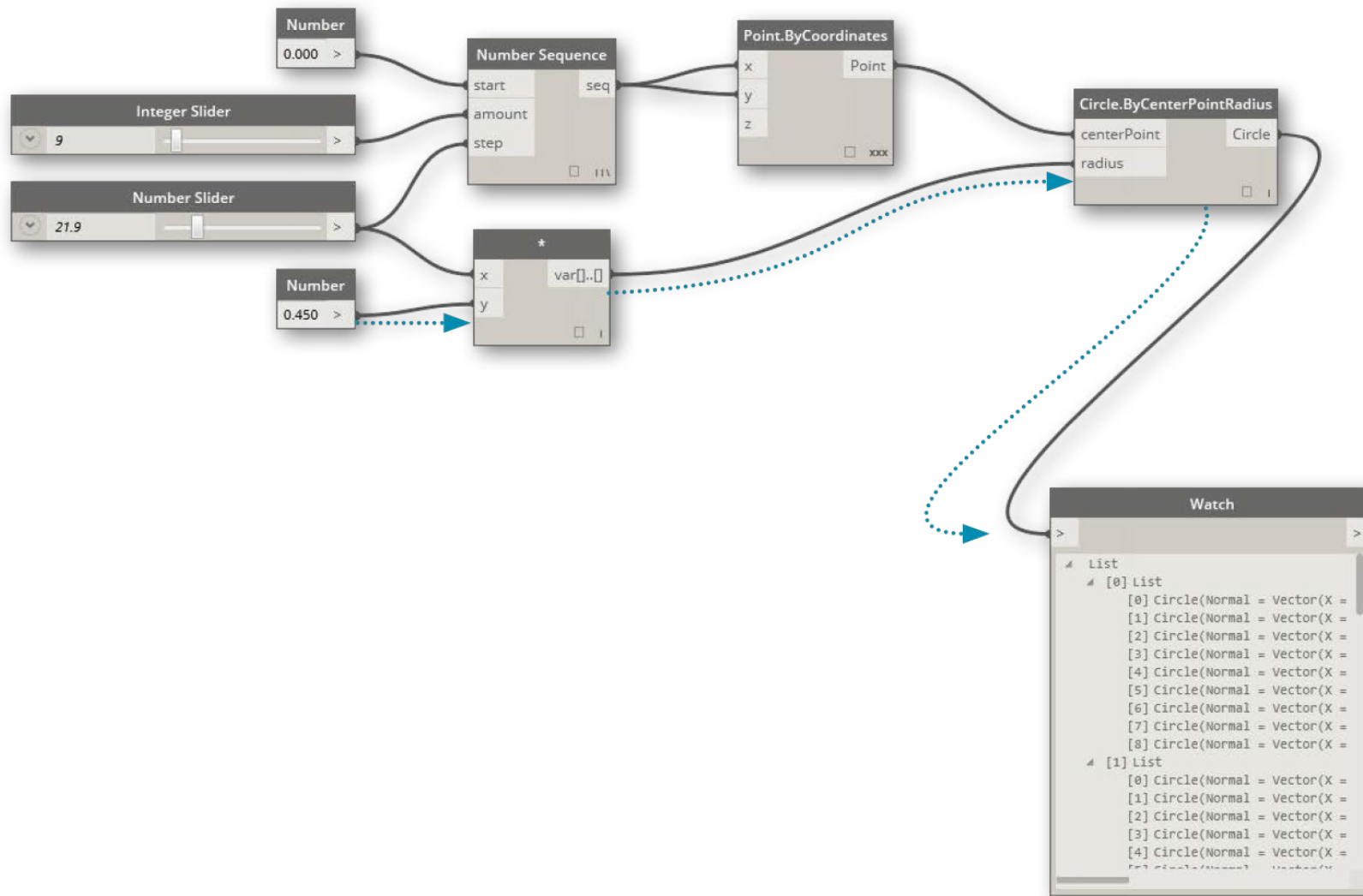
nodes



- 1) Name of the Node with a Category.Name naming convention
- 2) Main body of the Node
- 3) Ports (In and Out)
- 4) Data Preview – (hover over it)
- 5) Lacing Icon



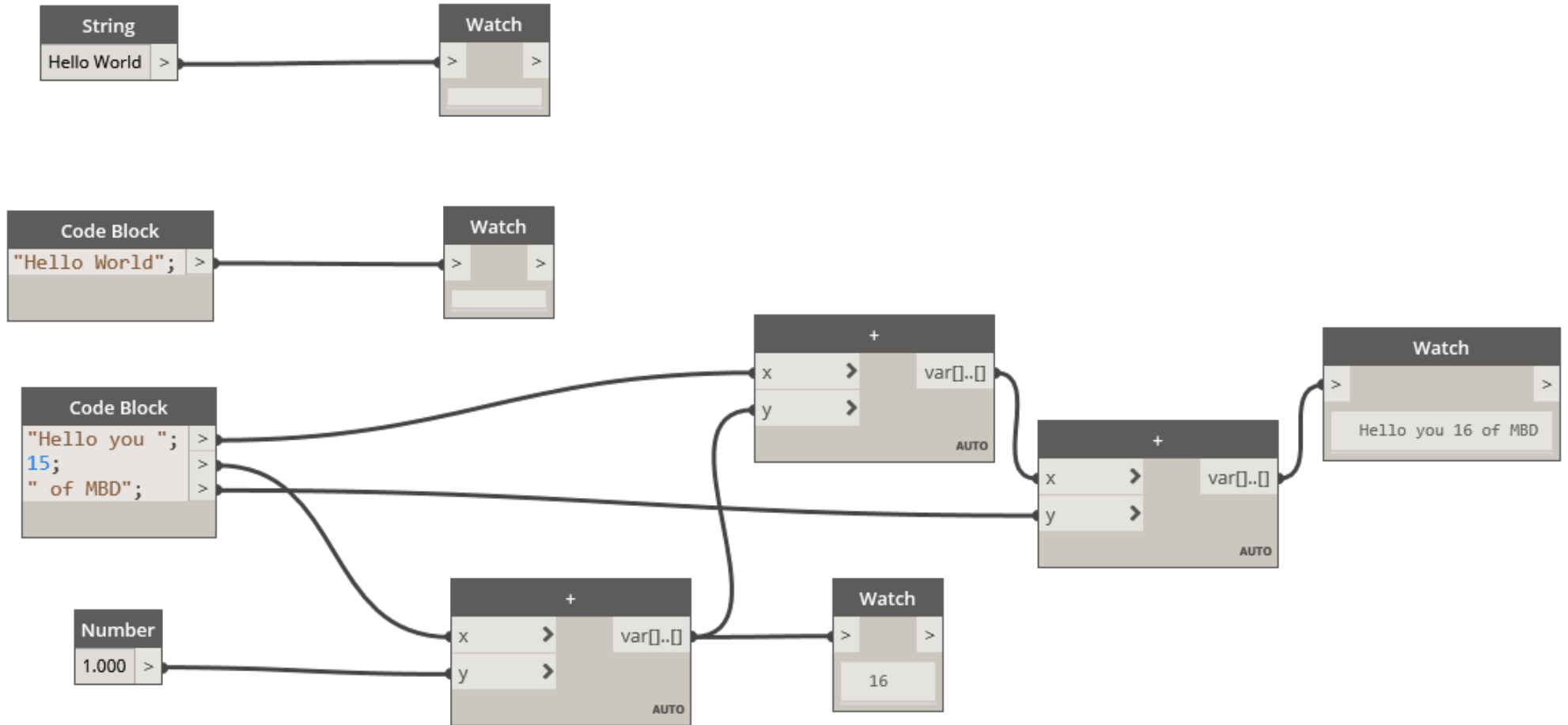
visual program



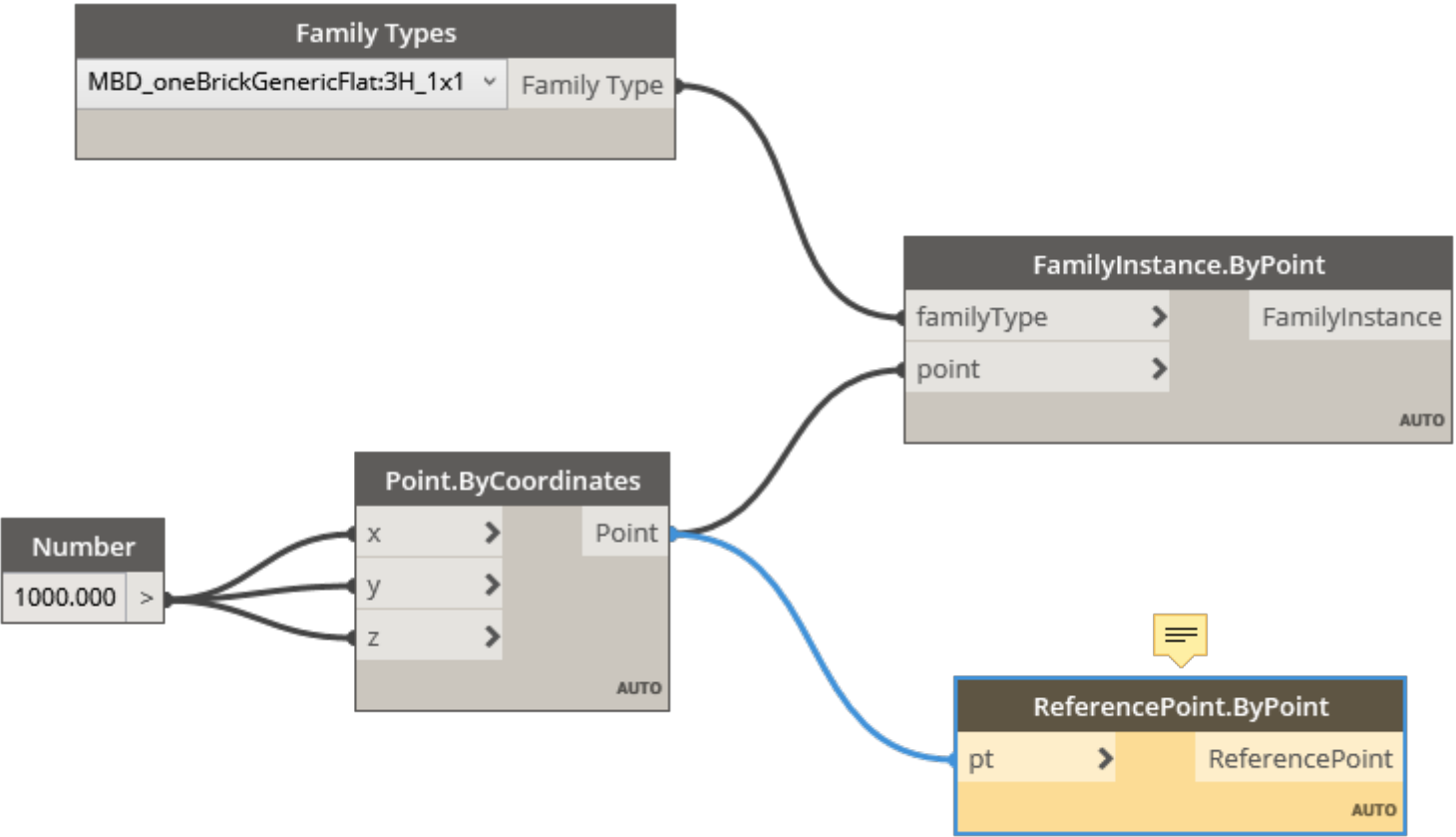
textual instructions

- `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);`

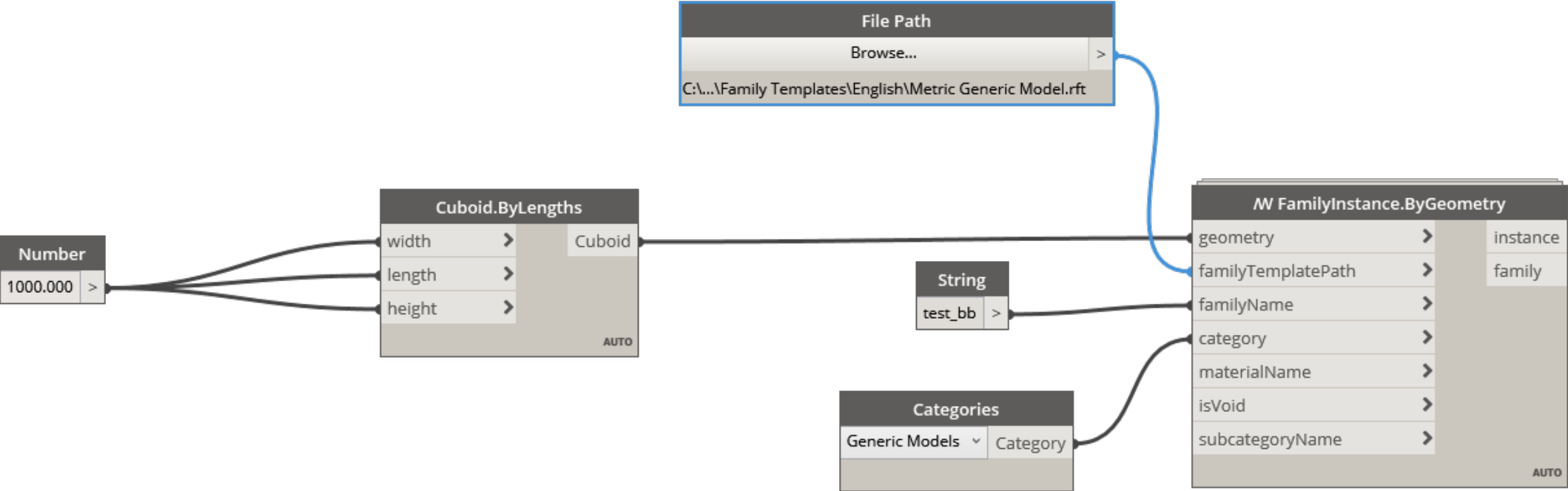
Hello World



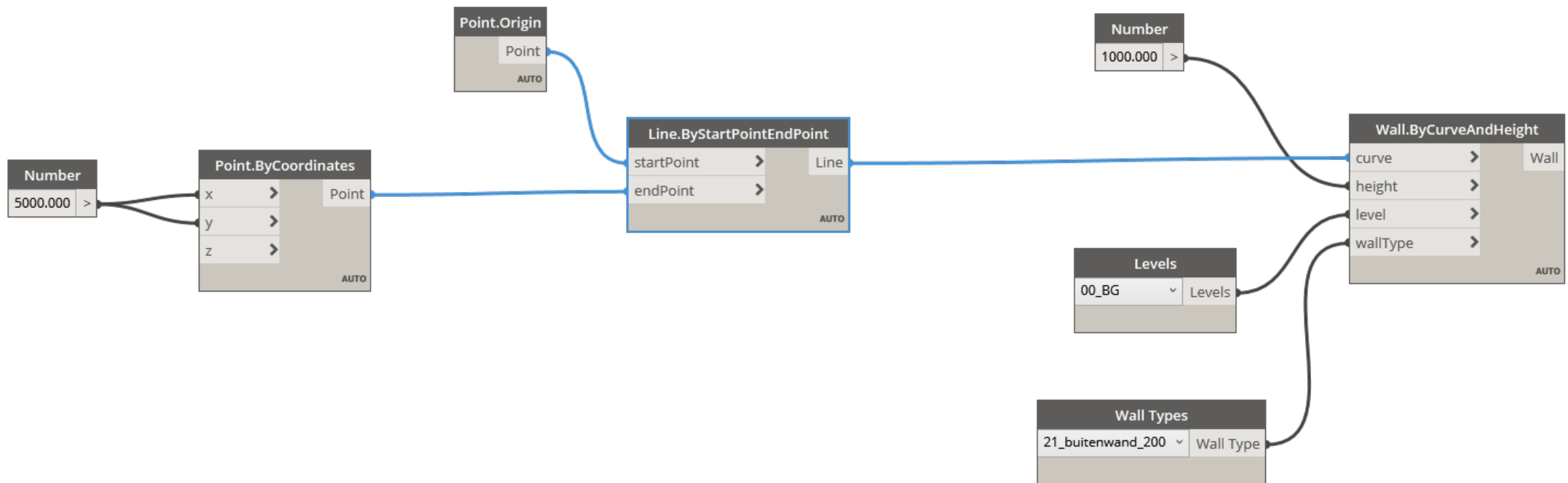
Points



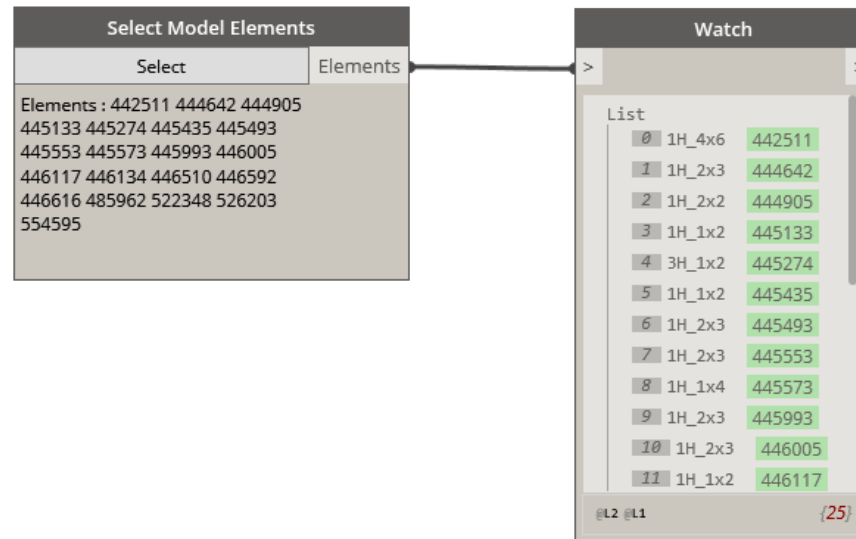
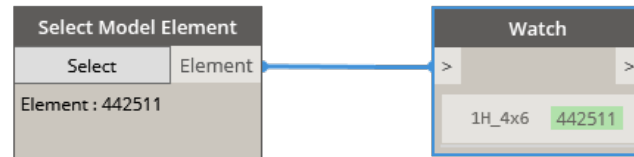
CubeToRevit



WallByCurve

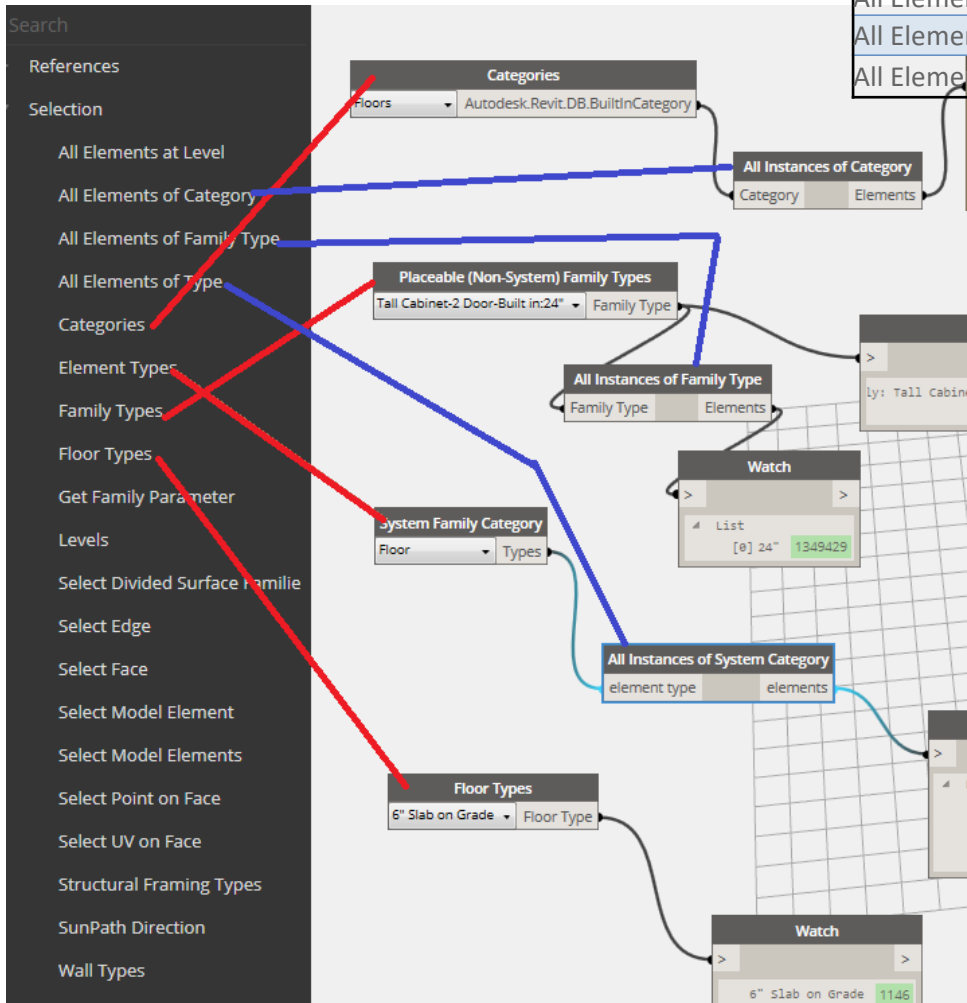


Make your selection...

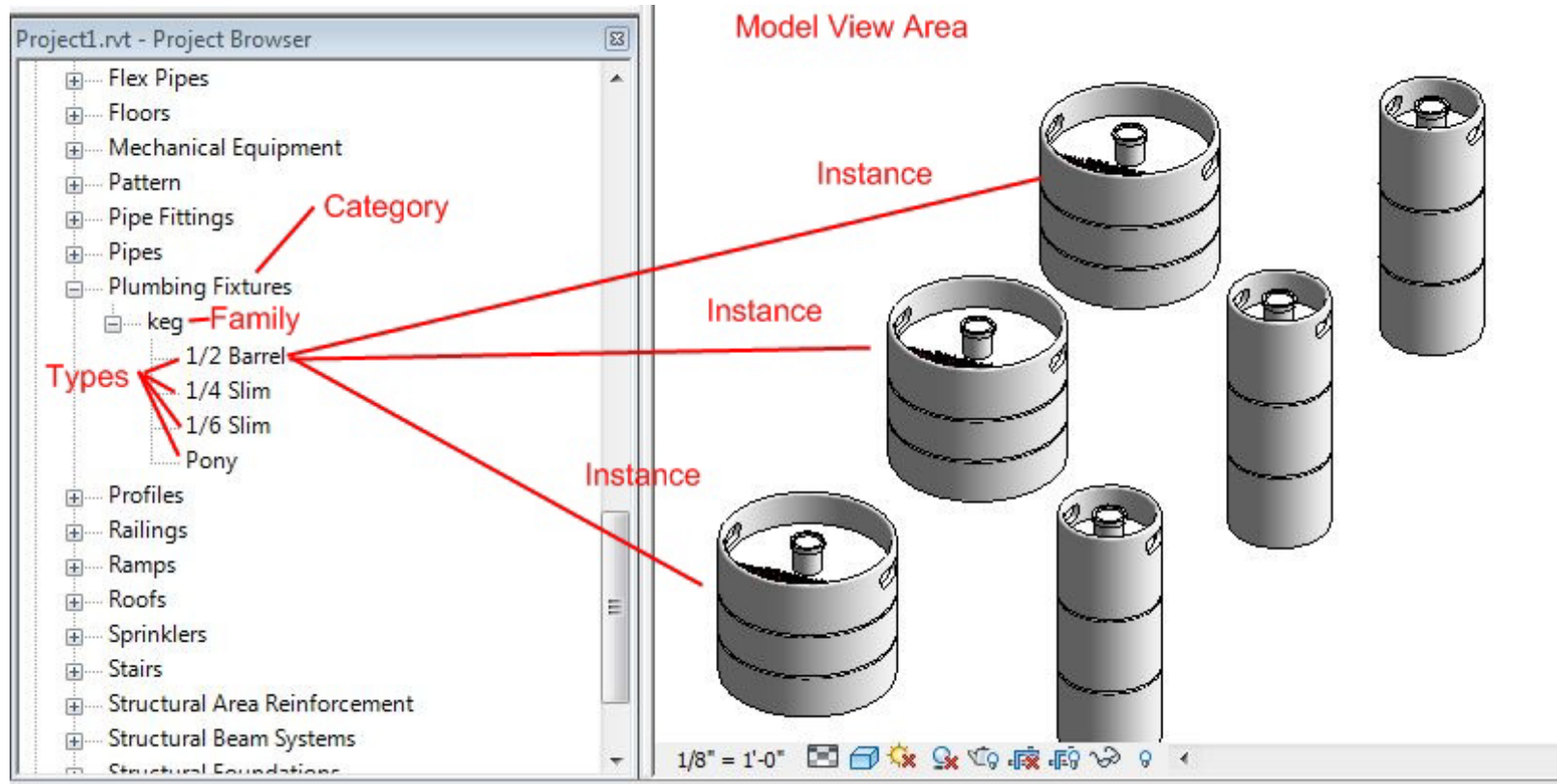


...from nodes

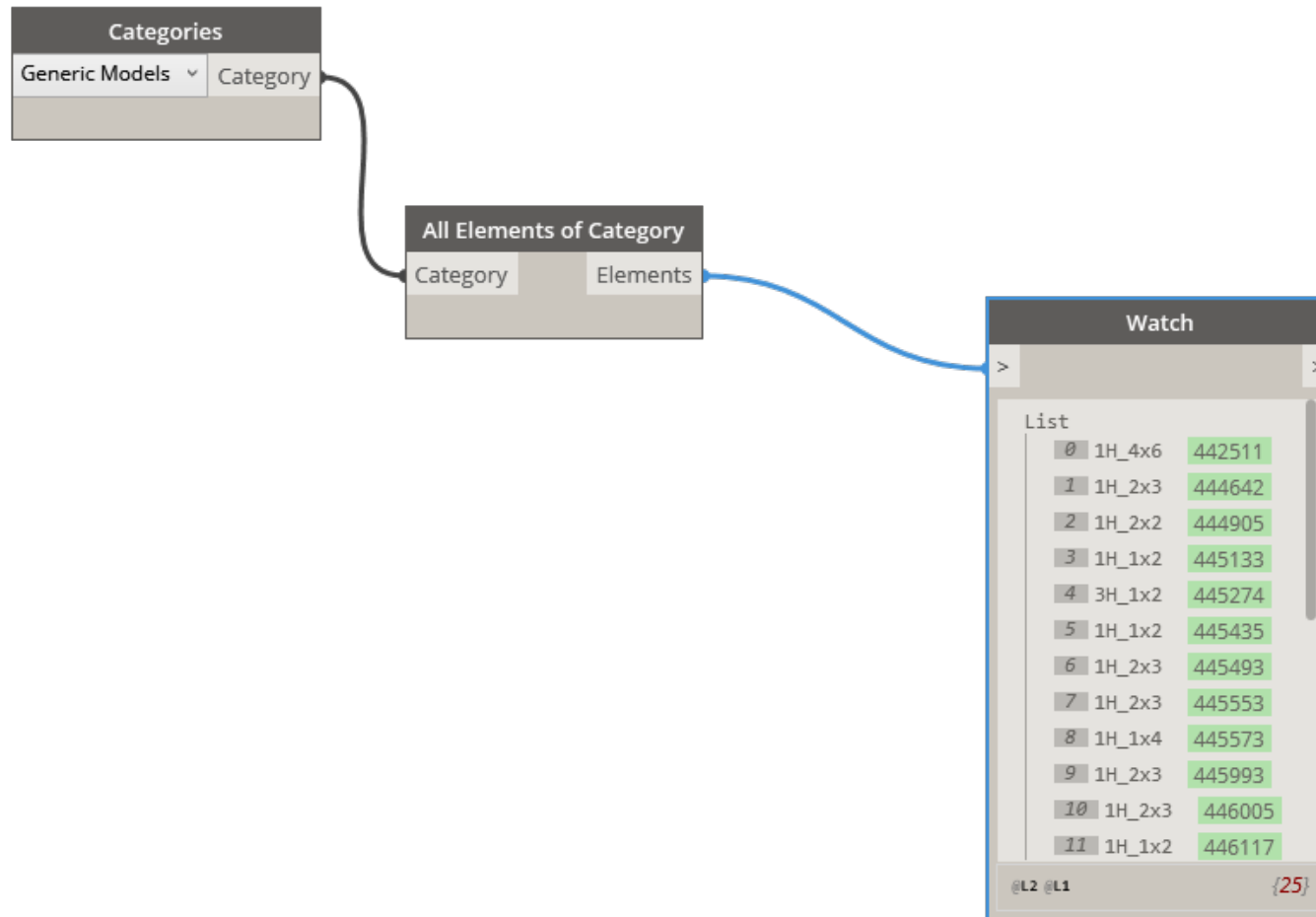
Current Node Name	What they Represent
Family Types	Placeable Family Types
Element Types	System Family Categories
All Elements of Family Type	All Instances of Family Type
All Elements of Type	All Instances of System Family Category
All Elements of Category	All Instances of Category



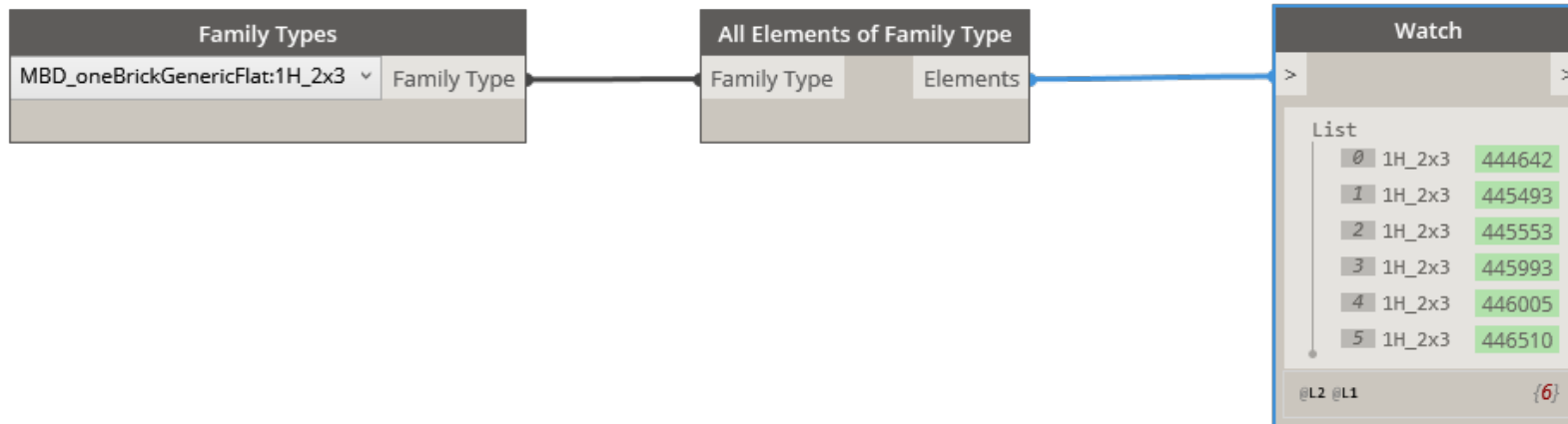
category, family, type, instance



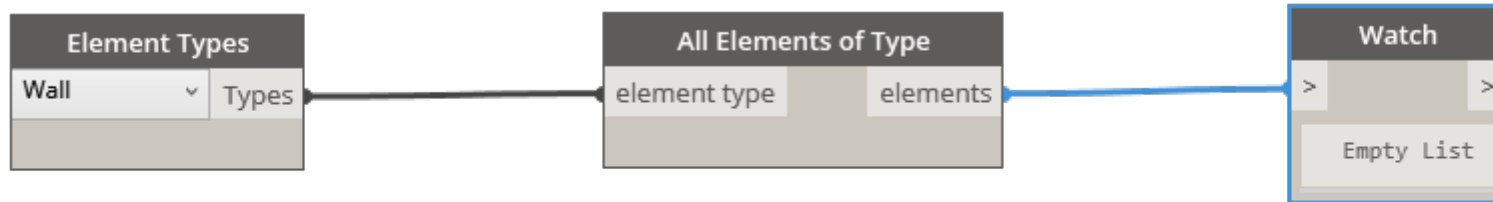
Categories



FamilyTypes

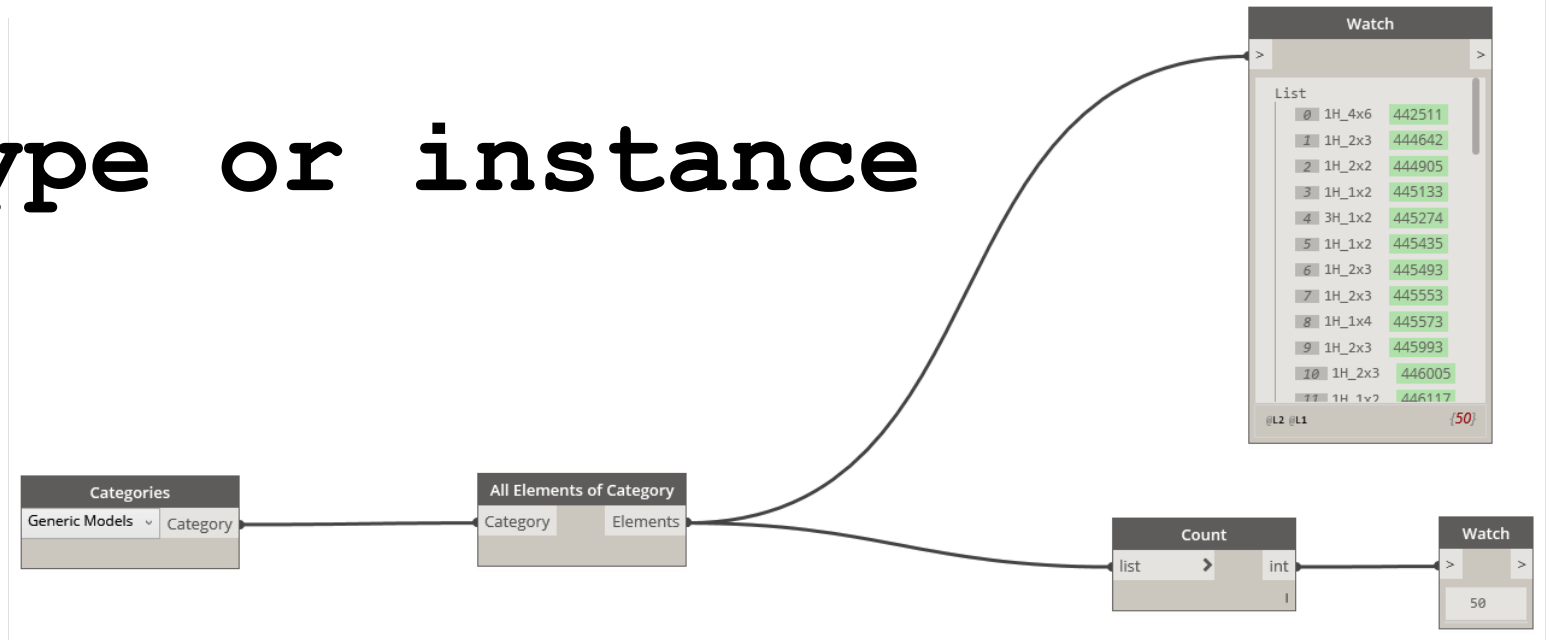


SystemFamilies

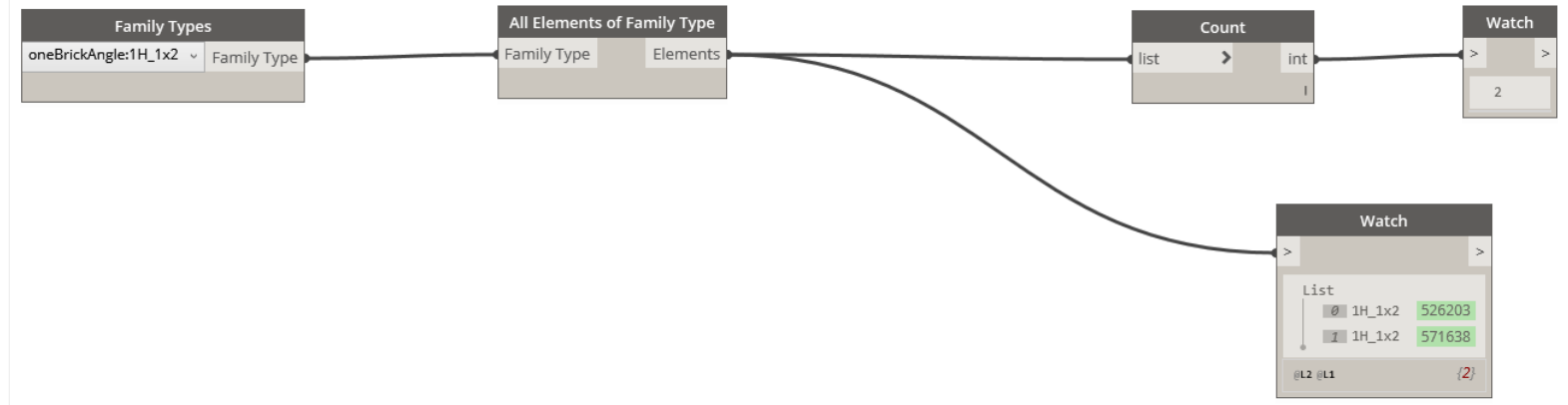


get type or instance

Get type:

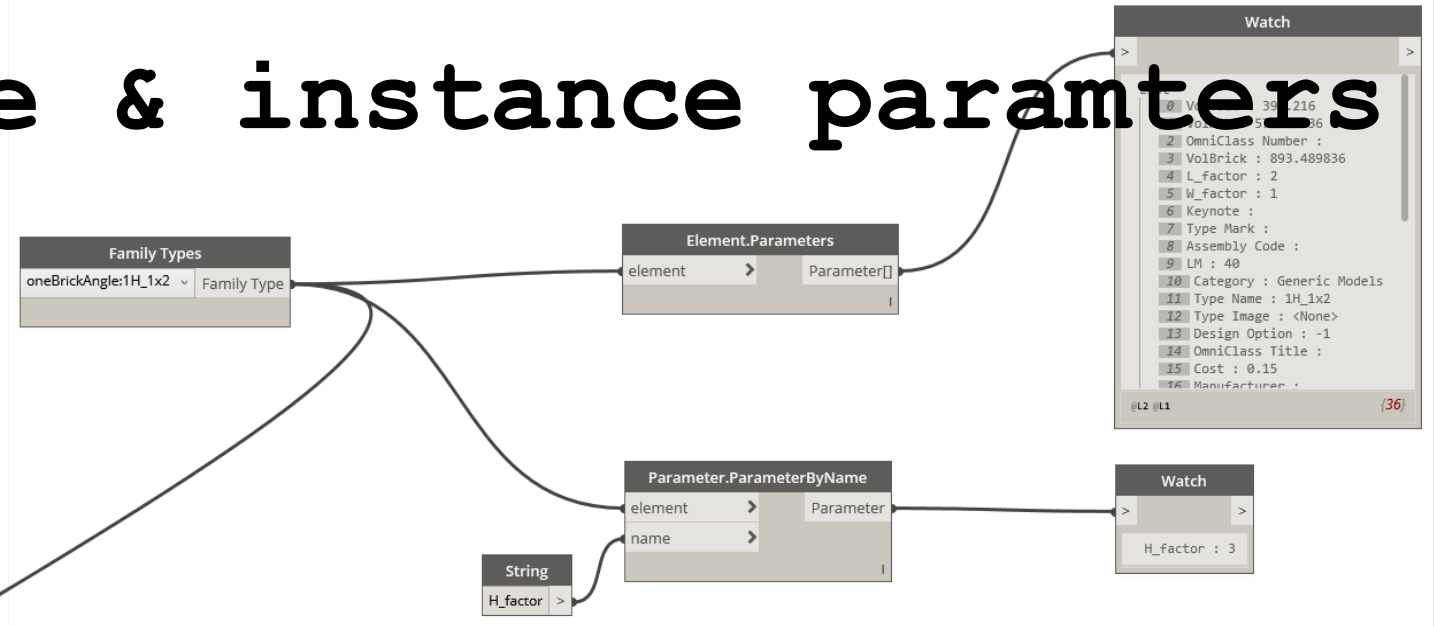


Get instance:

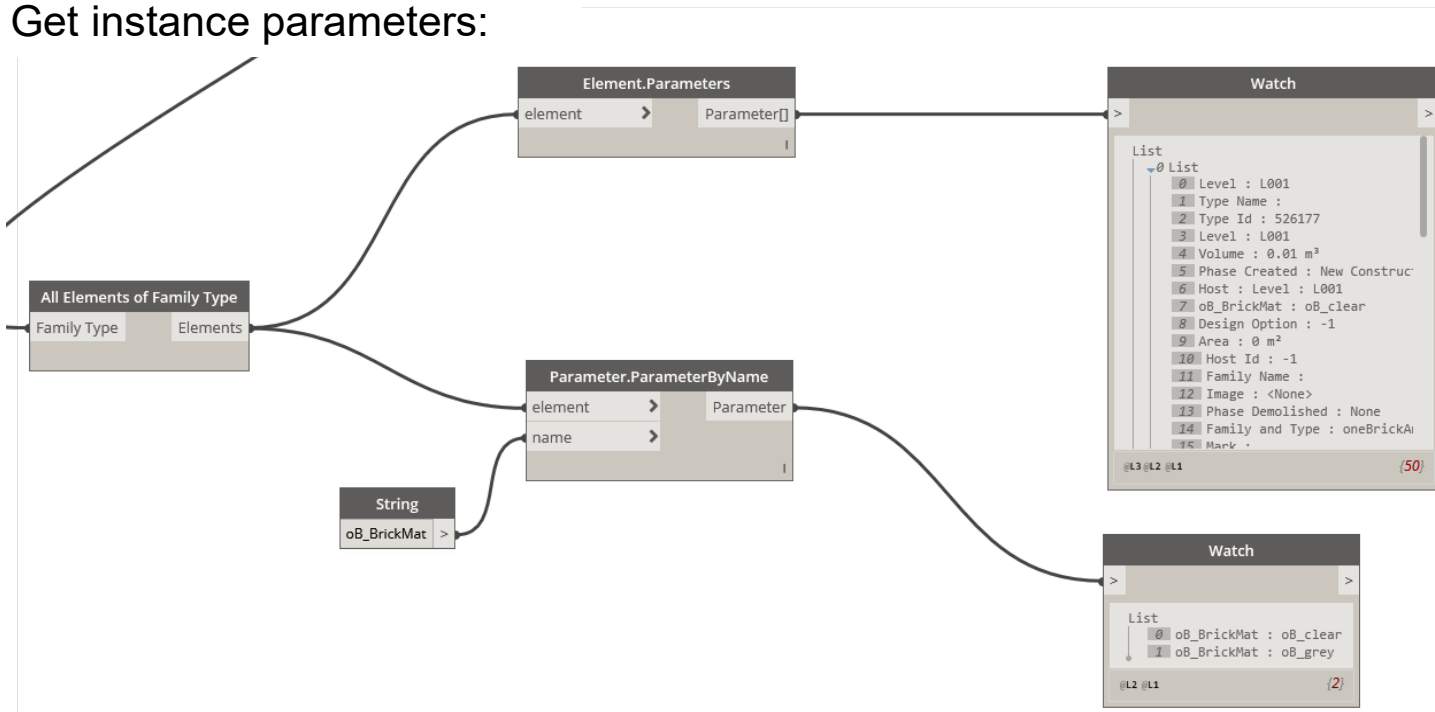


get type & instance parameters

Get type parameters:

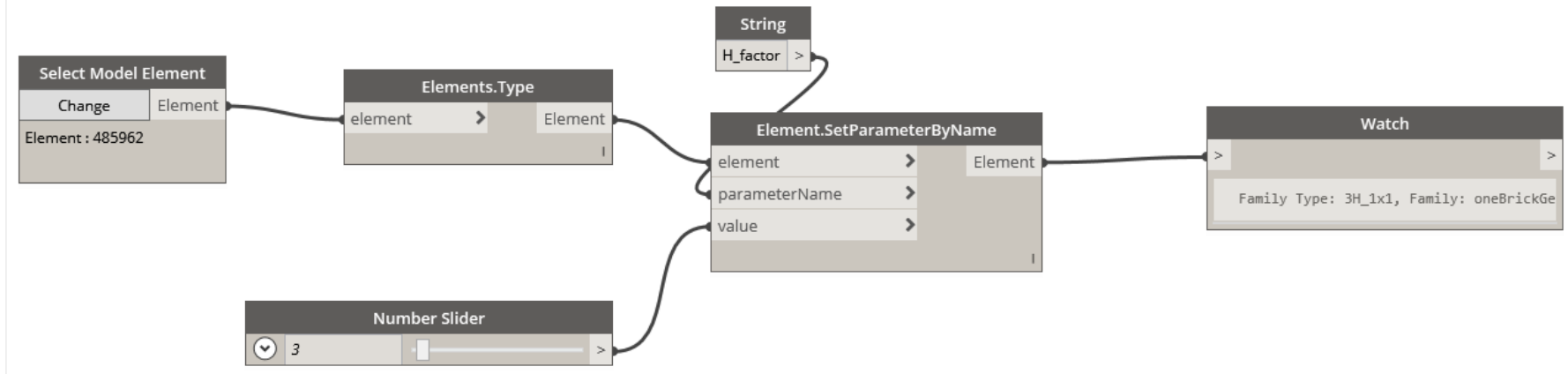


Get instance parameters:

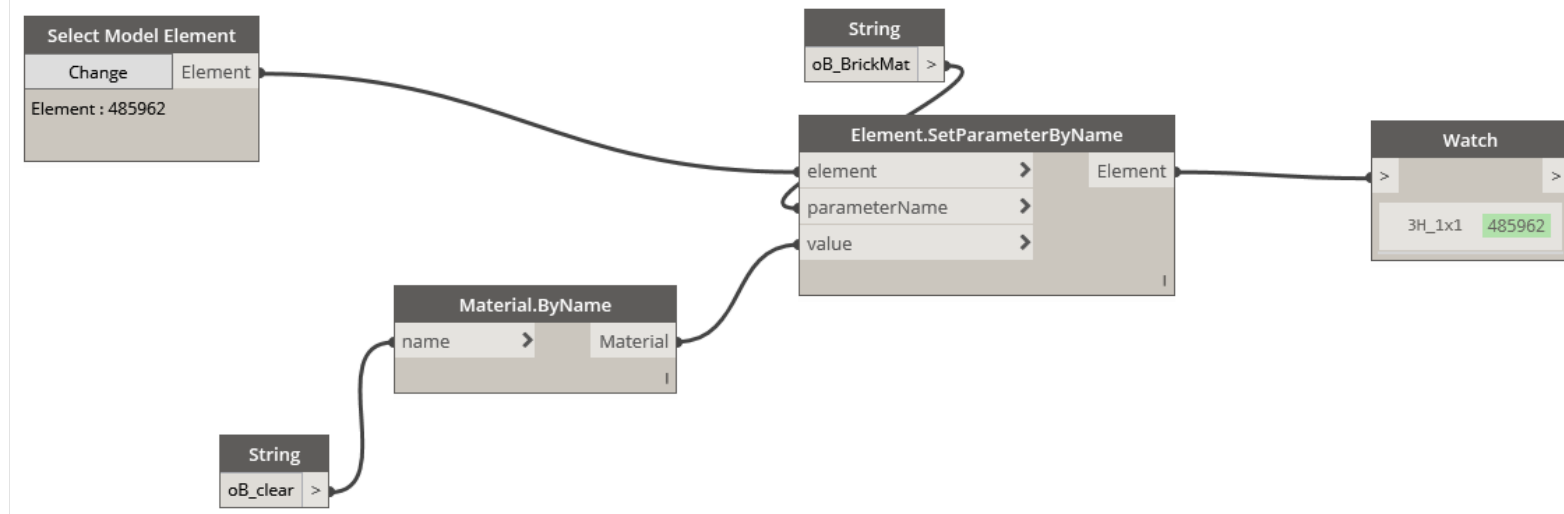


set instance paramters

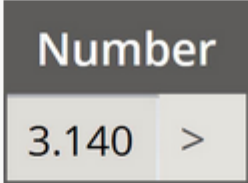

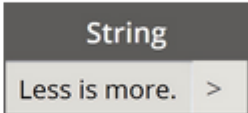

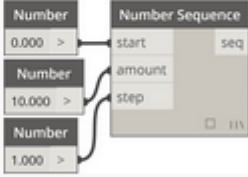
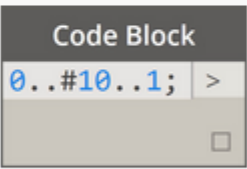
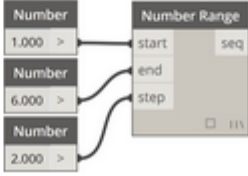
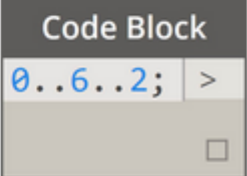
Set Type Parameter





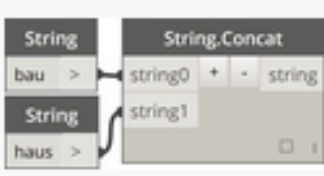
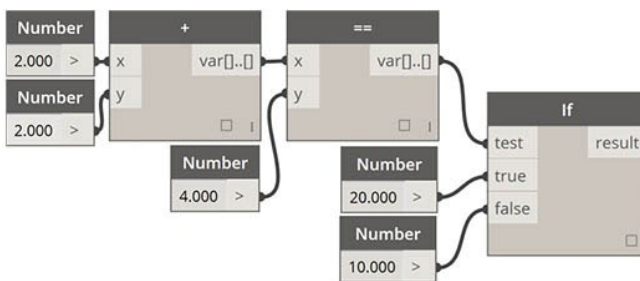
Set Instance Parameter



code block 01

Numbers		
Strings		
Sequences		
Ranges		

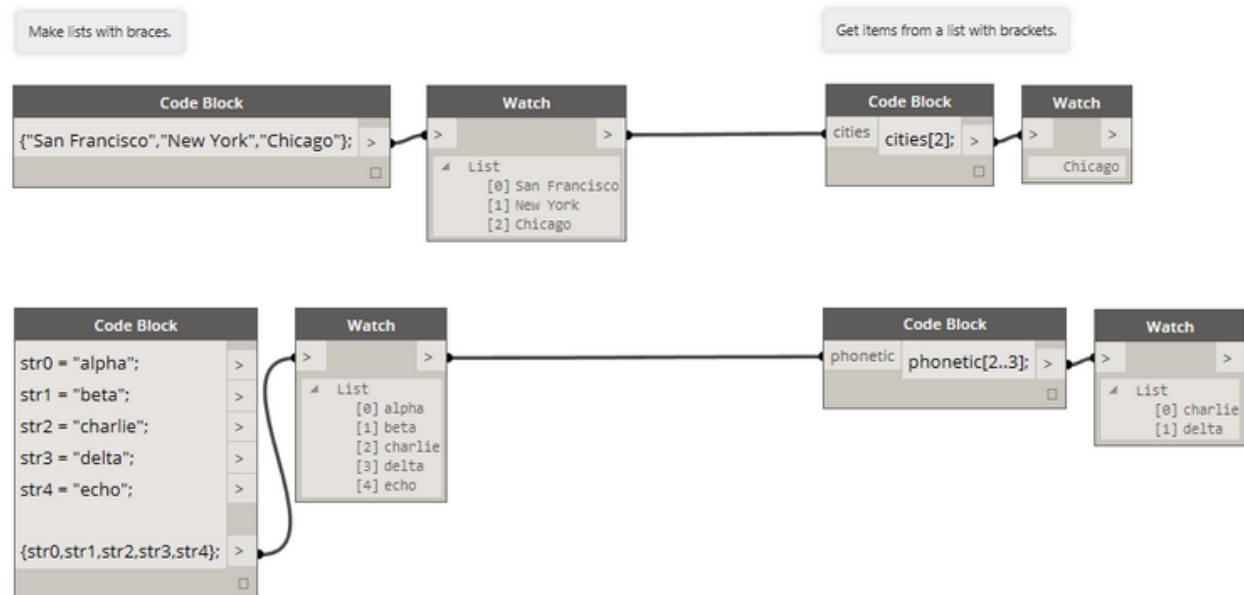
code block 02

Get Item at Index	 <p>A diagram showing the <code>List.GetItemAtIndex</code> function. It has two inputs: <code>list</code> (Number) with value <code>1,000</code> and <code>index</code> (Number) with value <code>1,000</code>. The output is <code>var[]..[]</code>.</p>	<pre>Code Block myList myList[1]; ></pre>
Create List	 <p>A diagram showing the <code>List.Create</code> function. It has three inputs: <code>index0</code> (Number) with value <code>0,000</code>, <code>index1</code> (Number) with value <code>3,000</code>, and <code>index2</code> (String) with value <code>dataString</code>. The output is <code>list</code>.</p>	<pre>Code Block {0,3,"dataString"}; ></pre>
Concatenate Strings	 <p>A diagram showing the <code>String.Concat</code> function. It has two inputs: <code>string0</code> (String) with value <code>bau</code> and <code>string1</code> (String) with value <code>haus</code>. The output is <code>string</code>.</p>	<pre>Code Block "bau"+"haus"; ></pre>
Conditional Statements	 <p>A diagram showing conditional statements. It starts with an addition: <code>x</code> (Number) with value <code>2,000</code> and <code>y</code> (Number) with value <code>2,000</code> are added to produce <code>4,000</code>. This result is compared (<code>==</code>) to <code>20,000</code>. The result of the comparison is used in an <code>If</code> statement. The <code>test</code> input of the <code>If</code> statement is the comparison result, and the <code>result</code> output is <code>true</code> if the test is successful and <code>false</code> otherwise. The <code>test</code> input is also shown with a value of <code>10,000</code>.</p>	<pre>Code Block 2+2==4?20:10; ></pre>

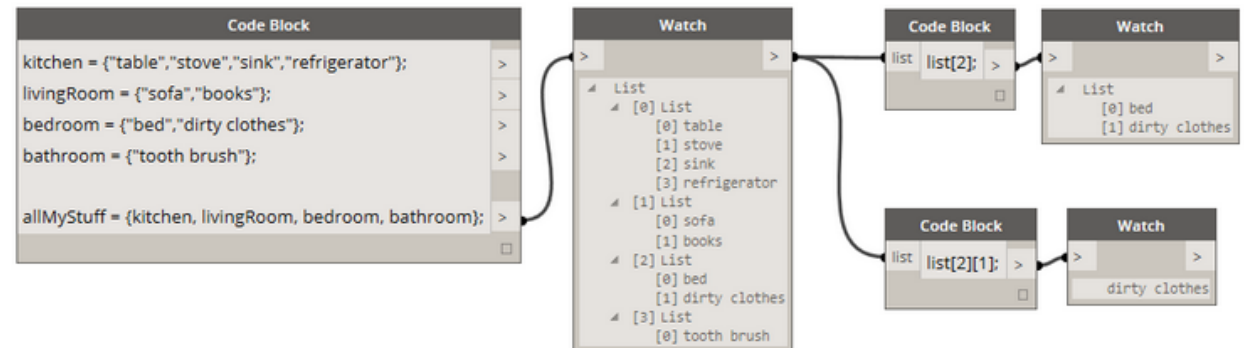
lists

Make lists and get items from a list

Make lists with braces (a.k.a. "curly brackets"). Get items from a list with brackets (a.k.a. "square brackets").



Working with nested lists is just a variation on the theme.

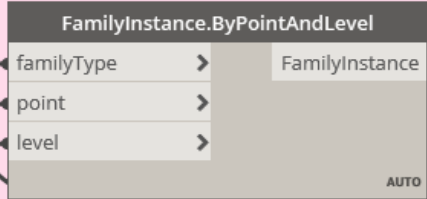


Organize!

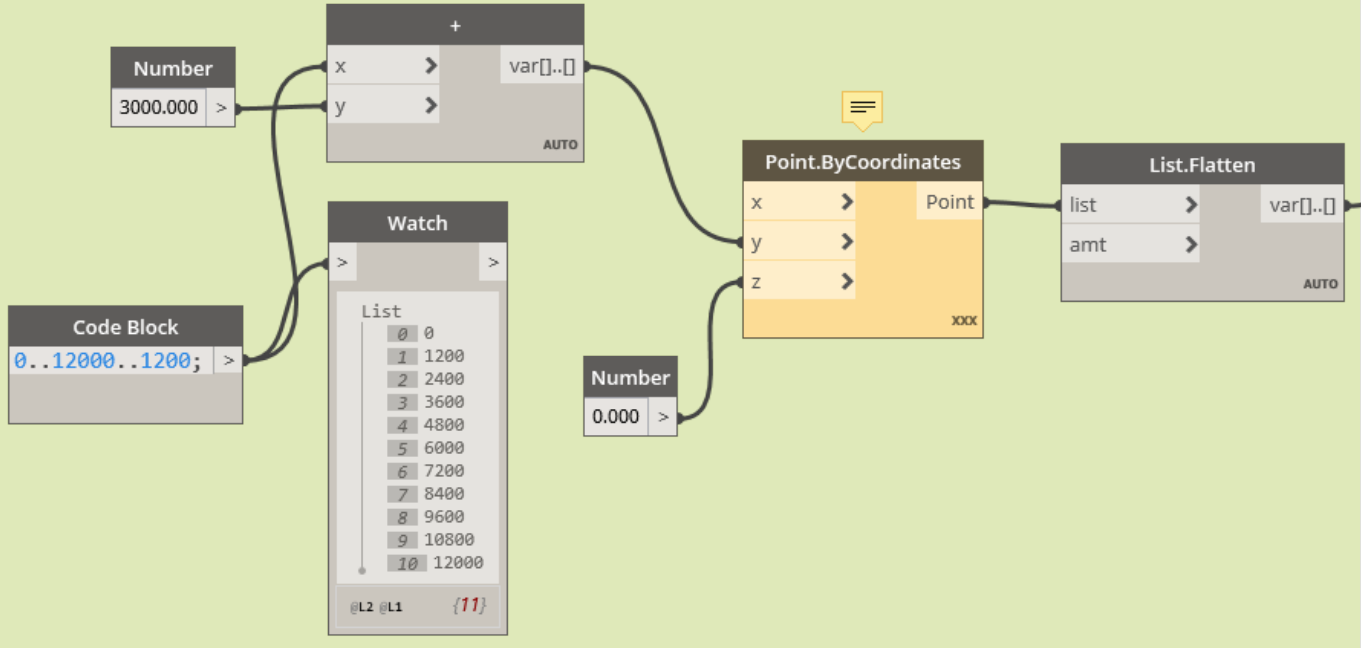
Select all placed Family Types



Place bricks in model on points



Make grid of points



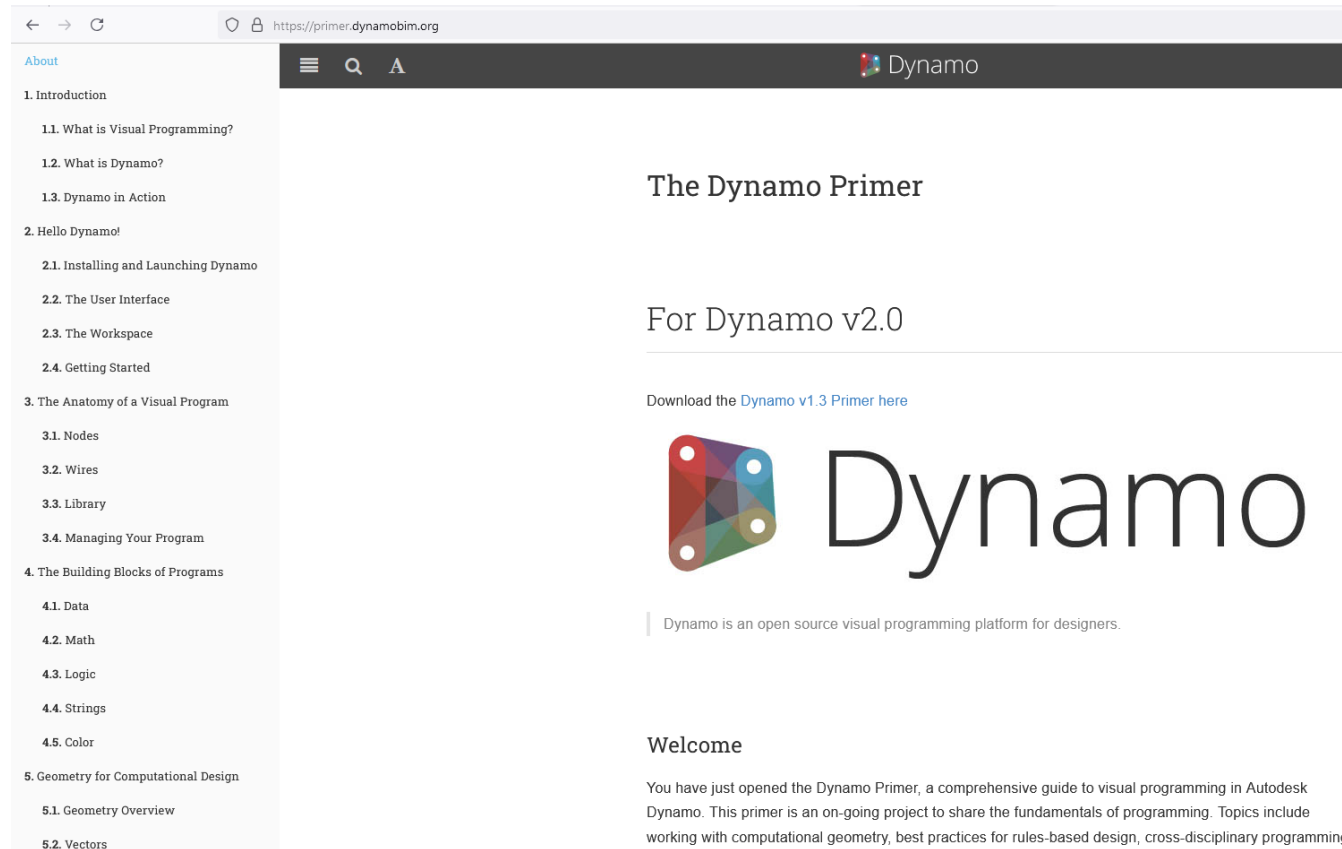
remember



yearMonthDay_Project_Title_Author_Version.xxx

Example: **20200313_MBD_PlaceBrick_V01.dyn**

Link to the primer (online dynamo tutorial)...



The screenshot shows a web browser window with the URL <https://primer.dynamobim.org>. The page features a dark header with the 'Dynamo' logo. On the left, a navigation menu lists sections from '1. Introduction' to '5.2. Vectors'. The main content area is titled 'The Dynamo Primer' and 'For Dynamo v2.0'. It includes a link to 'Download the Dynamo v1.3 Primer here', the Dynamo logo, and a description: 'Dynamo is an open source visual programming platform for designers.' Below this is a 'Welcome' section with a paragraph: 'You have just opened the Dynamo Primer, a comprehensive guide to visual programming in Autodesk Dynamo. This primer is an on-going project to share the fundamentals of programming. Topics include working with computational geometry, best practices for rules-based design, cross-disciplinary programming'.

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

...