B


Dynamo

TRAINING OVERVIEW

Module 01

Dyn_01_Getting started
Dyn_02_Arithmetic and Logic Operators
Dyn_03_Strings and other Data Types

Module 02

Dyn_04_List Management
Dyn_05_List Management_Example
Dyn_06_DesignScript and codeblocks

Module 03

Dyn_07_Working with Revit Data
Dyn_08_Revit Data Management_Example01
Dyn_09_Revit Data Management_Example02
Dyn_10_Revit to Excel
Dyn_11_Excel to Revit

Module 04

Dyn_12_Create Sheets
Dyn_13_Create Sheets from Excel file
Dyn_14_Room numbering
Dyn_15_Element Grid location

TRAINING OVERVIEW

Module 01

Dyn_01_Getting started
Dyn_02_Arithmetic and Logic Operators
Dyn_03_Strings and other Data Types

Understanding Dynamo and Visual Scripting

Module 02

Dyn_04_List Management
Dyn_05_List Management_Example
Dyn_06_DesignScript and codeblocks

List and Data management

Module 03

Dyn_07_Working with Revit Data
Dyn_08_Revit Data Management_Example01
Dyn_09_Revit Data Management_Example02
Dyn_10_Revit to Excel
Dyn_11_Excel to Revit

Working with Revit data

Module 04

Dyn_12_Create Sheets
Dyn_13_Create Sheets from Excel file
Dyn_14_Room numbering
Dyn_15_Element Grid location

Use cases

TRAINING FILES

1. Learn the basics about Visual Programming

2. Understand Data Types

3. Dynamo programming mindset

4. Troubleshoot existing scripts (errors)

5. Update KAAN standard scripts

6. Create your own scripts

TRAINING FILES

N:\02 BIM\02_training\03_BIM training\Dynamo

DYNAMO RESOURCES

- **Dynamo primer**
https://primer2.dynamobim.org/

- **Dynamo Github**
https://github.com/DynamoDS/Dynamo

- **Dynamo forums**
https://forum.dynamobim.com/

- **Github where training scripts can be found**
https://github.com/JavierCuartero/Dynamo_Training


Dynamo

?

Visual Programming

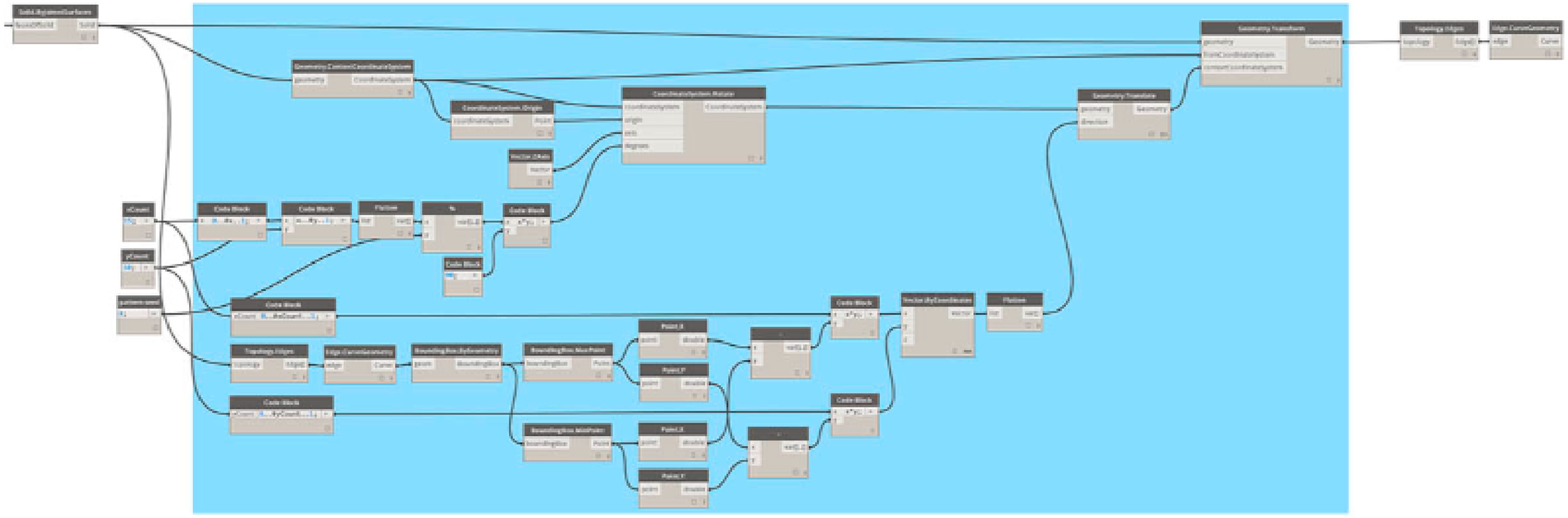
CODING

Python Script

```python
# Enable Python support and load DesignScript library
import clr
clr.AddReference('ProtoGeometry')
from Autodesk.DesignScript.Geometry import *

# The inputs to this node will be stored as a list in the IN variables.
#The solid module to be arrayed
solid = IN[0]
#A number that determines which rotation pattern to use
seed = IN[1]
#The number of solids to array in the X and Y axes
xCount = IN[2]
yCount = IN[3]

#Create an empty list for the arrayed solids
solids = []
# Create an empty list for the edge curves
crvs = []

# Place your code below this line
#Loop through edges and append corresponding curve geometry to the list
for edge in solid.Edges:
    crvs.append(edge.CurveGeometry)
#Get the bounding box of the curves
bbox = BoundingBox.ByGeometry(crvs)

#Get the X and Y translation distance based on the bounding box
yDist = bbox.MaxPoint.Y-bbox.MinPoint.Y
xDist = bbox.MaxPoint.X-bbox.MinPoint.X


# Assign your output to the OUT variable.
OUT = solids
```

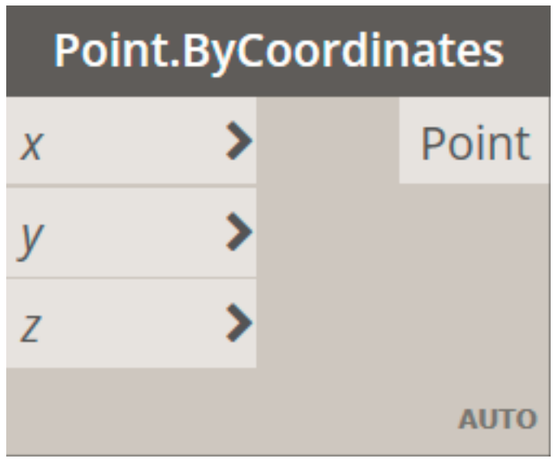
Run IronPython2

VISUAL PROGRAMMING

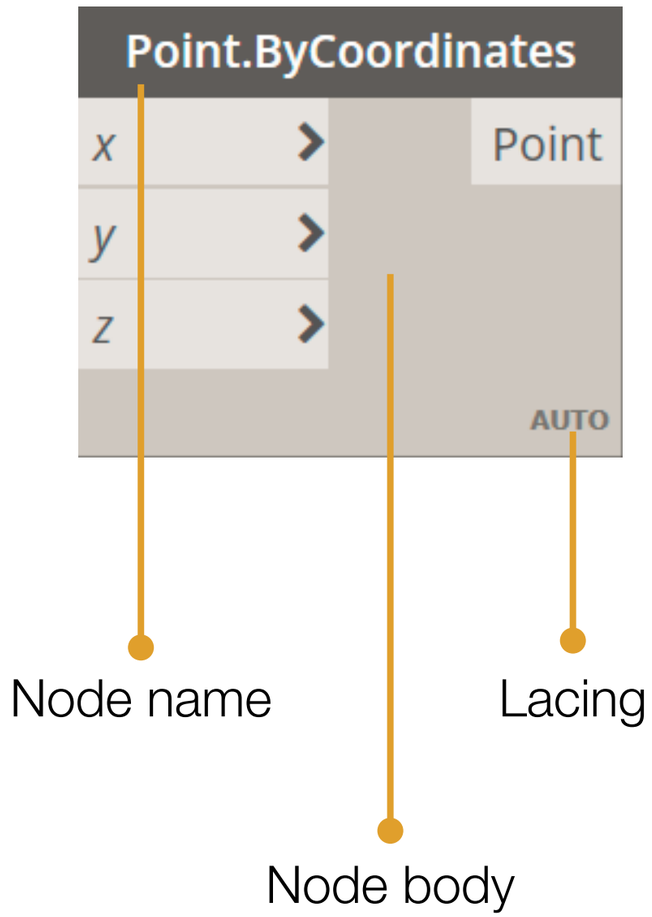


Nodes

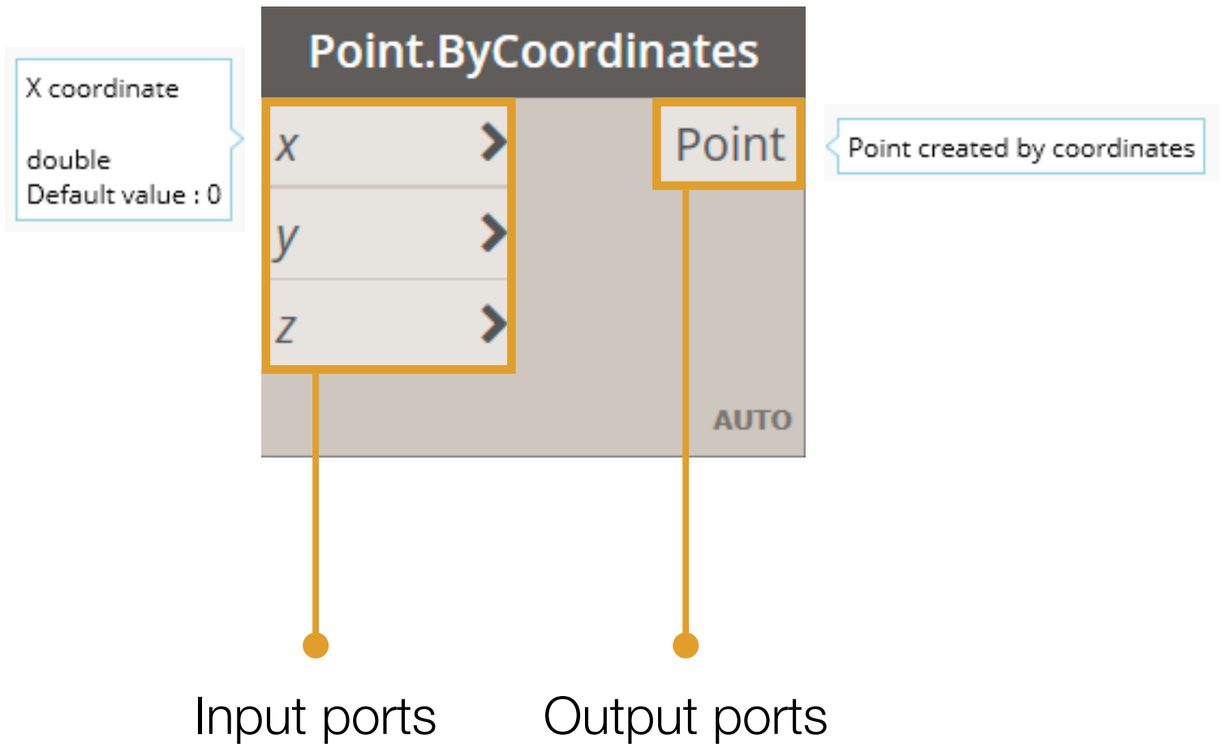
NODES



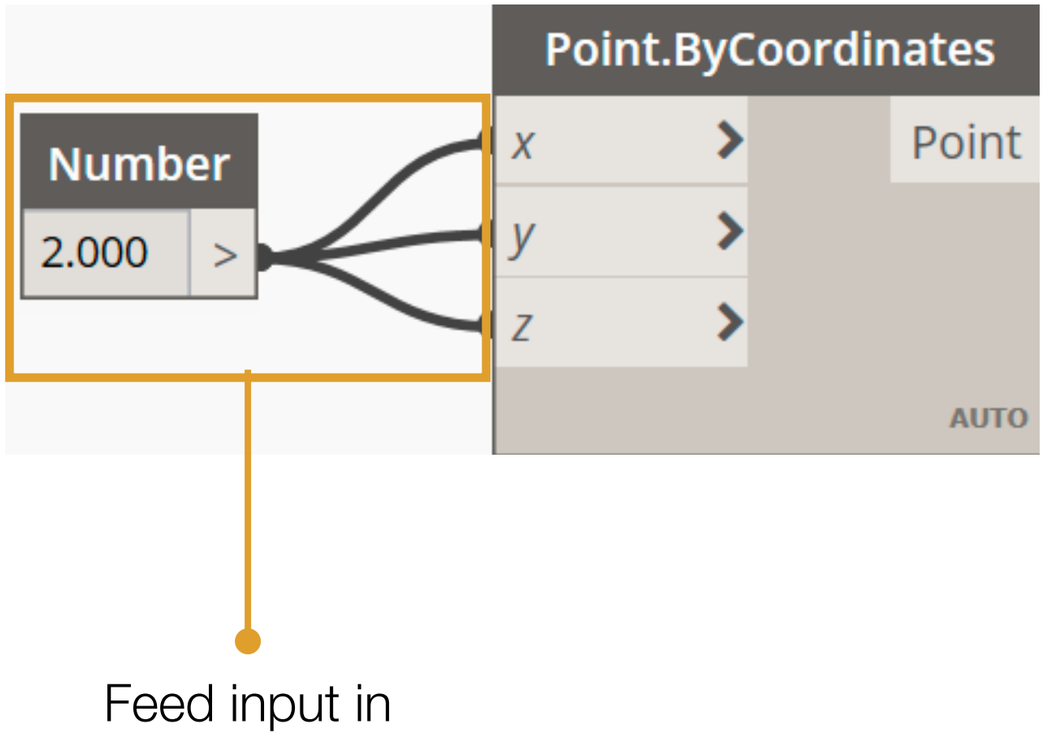
NODES


Point.ByCoordinates
x
y
z
Point
AUTO
Node name
Node body
Lacing

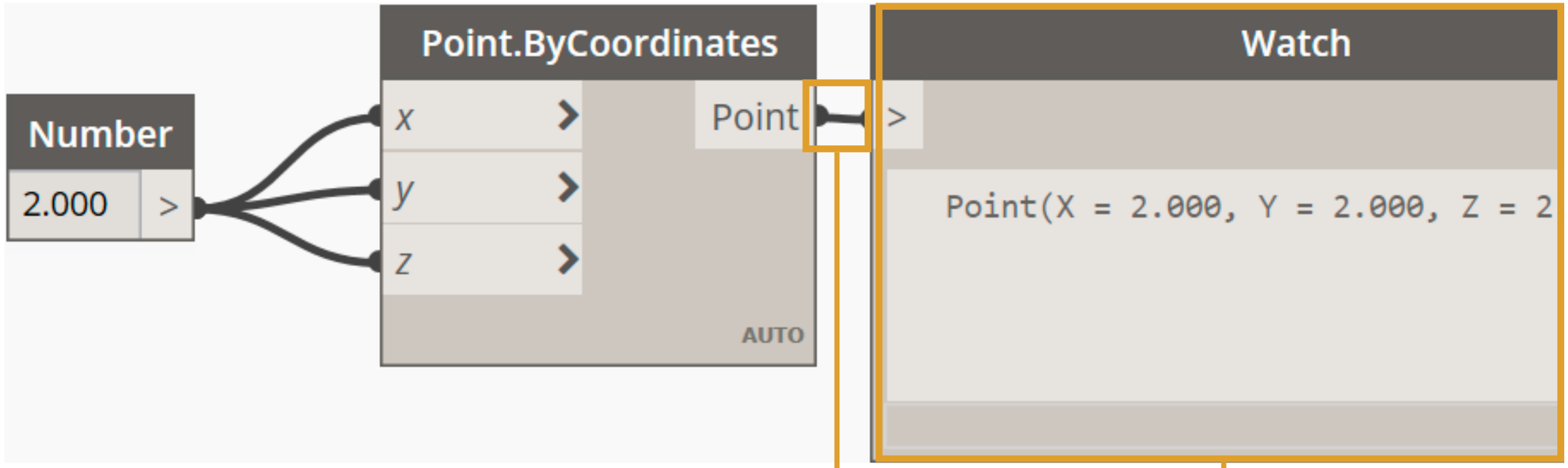
NODES: INPUT & OUTPUT



CONNECTING NODES



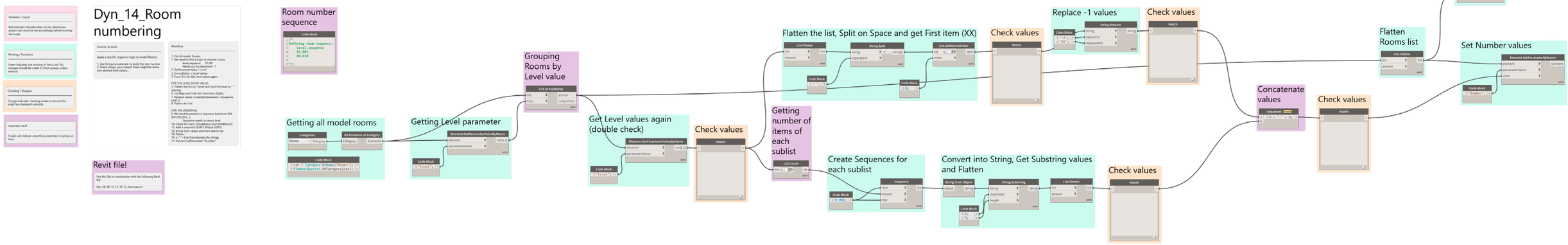
CONNECTING NODES



Connect output to Watch node

Node output:
point created at 2,2,2

SCRIPT EXAMPLE



Let’s jump into Dynamo!