

Session 2.2 - DesignScript: A robust dive into the language underlying Dynamo

Tip 1: Name your DesignScript **variables** logically! Not only will others understand easier, but you will when going back to old code.

Code Block		
curvesList	good = curvesList;	>
var1	bad = var1;	>

Tip 2: Turn off '**Preview**' on the Code Block and use another code block to 'pass through' only items you want to visualise geometrically by using a *dummy variable*.

Code Block		
stuff	stuff;	>
morestuff	morestuff;	>
lotsmoreStuff	lotsmoreStuff;	>
	Point.ByCoordinates();	>

Code Block		
dummy	dummy;	>

Point(X = 0.000, Y = 0.000, Z = 0.000)

But name it something better than dummy!

Session 2.2 - DesignScript: A robust dive into the language underlying Dynamo

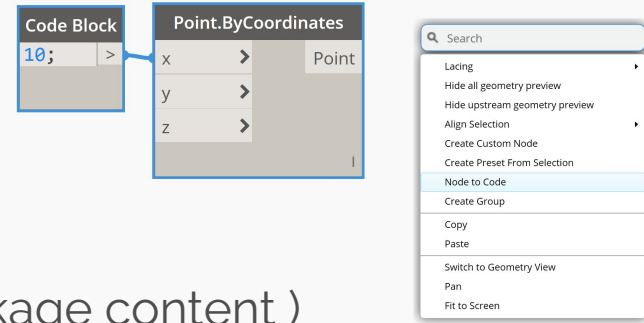
Tip 3: Only **nest** constructors when you do not need them further down the line. This will save on overall code lines.

```
Code Block
origin = Point.ByCoordinates();
baseline = Line.ByStartPointEndPoint(origin, Point.ByCoordinates(0, 10, 0));
nextLine = Line.ByStartPointEndPoint(Point.ByCoordinates(0, 10, 0),
Point.ByCoordinates(10, 20, 0));
// Hrm... should have created the second point as it's own variable to
// call in baseLine AND nextline
```

Tip 4: Dynamo can, over time, **fill up memory**. If working all day in Dynamo it pays to restart your session to *gain back speed*. If you are working in Dynamo-Revit then you need to restart Revit also as your Dynamo session is tethered.

Session 2.2 - DesignScript: A robust dive into the language underlying Dynamo

Tip 5: Use **Node2Code** lavishly to understand how a Node will translate to DesignScript (Select nodes and right click anywhere on the canvas).

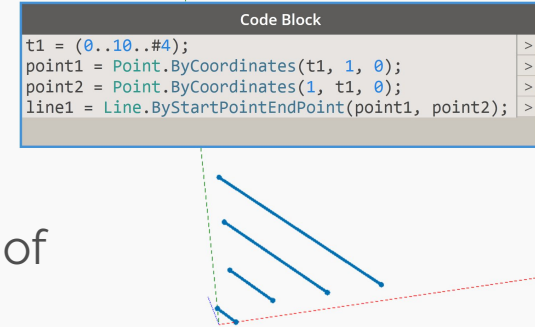


Tip 6: *Zero-touch* nodes (Custom C# package content) **can be called** inside of DesignScript. Python or Nodal based packages *cannot*.

Tip 7: Read up on the 1.X to 2.X language changes here: <https://github.com/DynamoDS/Dynamo/wiki/Dynamo-2.0-Language-Changes-Explained> so you don't pull your hair out when replicating old DesignScript logic.

Session 2.2 - DesignScript: A robust dive into the language underlying Dynamo

Tip 8: If you **Node2Code** a large Nodal graph - it will not be the most efficient (Not many nested components) or named in a legible manner. Better to use this feature as a learning mechanism and practise aptly naming your lines of code.



Tip 9: Get practised using DesignScript **range** and **sequence** - You won't go back to the Nodal form afterwards! Bring on more..ranges..#now !

Tip 10: DesignScript is an **easy-to-learn coding language** for designers as it's built from parts we conceptually understand: Lines, points, surfaces etc.

