Computational Fabrication

CS 491 and 591
Professor: Leah Buechley
https://handandmachine.org/classes/computational_fabrication/

Large Assignment 2 Posted

Due Tuesday 9/26 (one week from today)

Three parametric 3D printed vessels

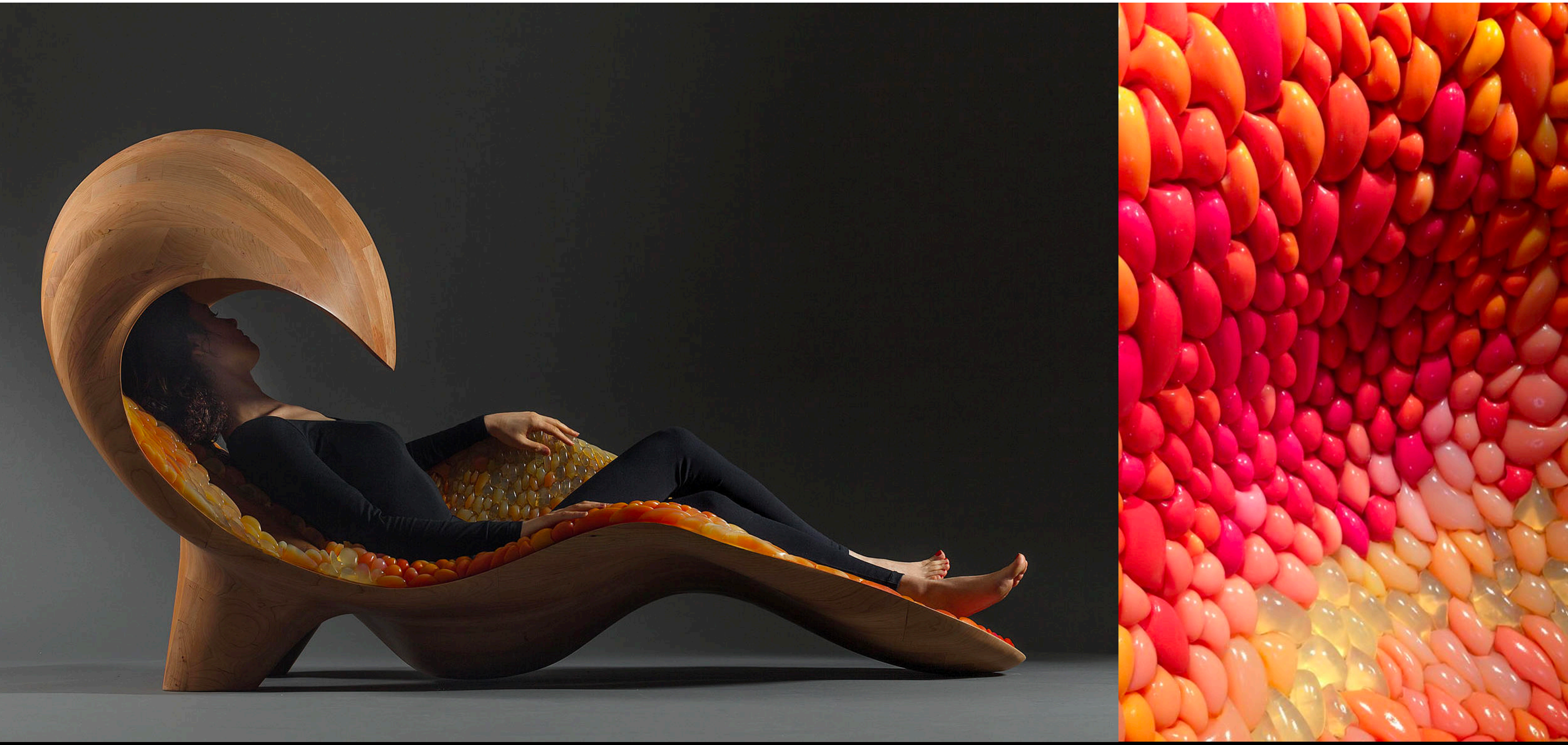
Comments and responses

Weekly Designer: Neri Oxman

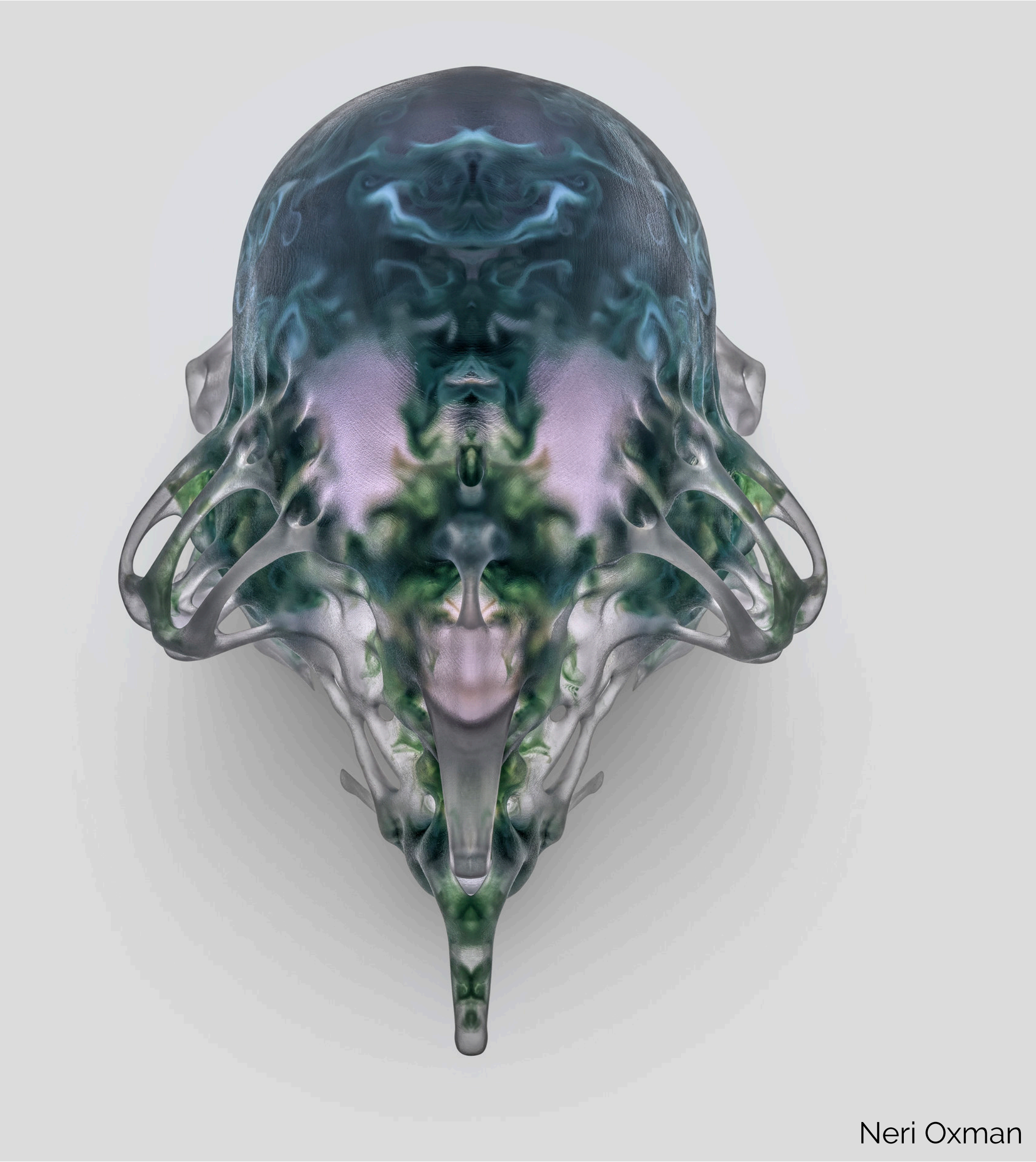
https://oxman.com/projects

https://www.media.mit.edu/groups/mediated-matter/archived-projects/

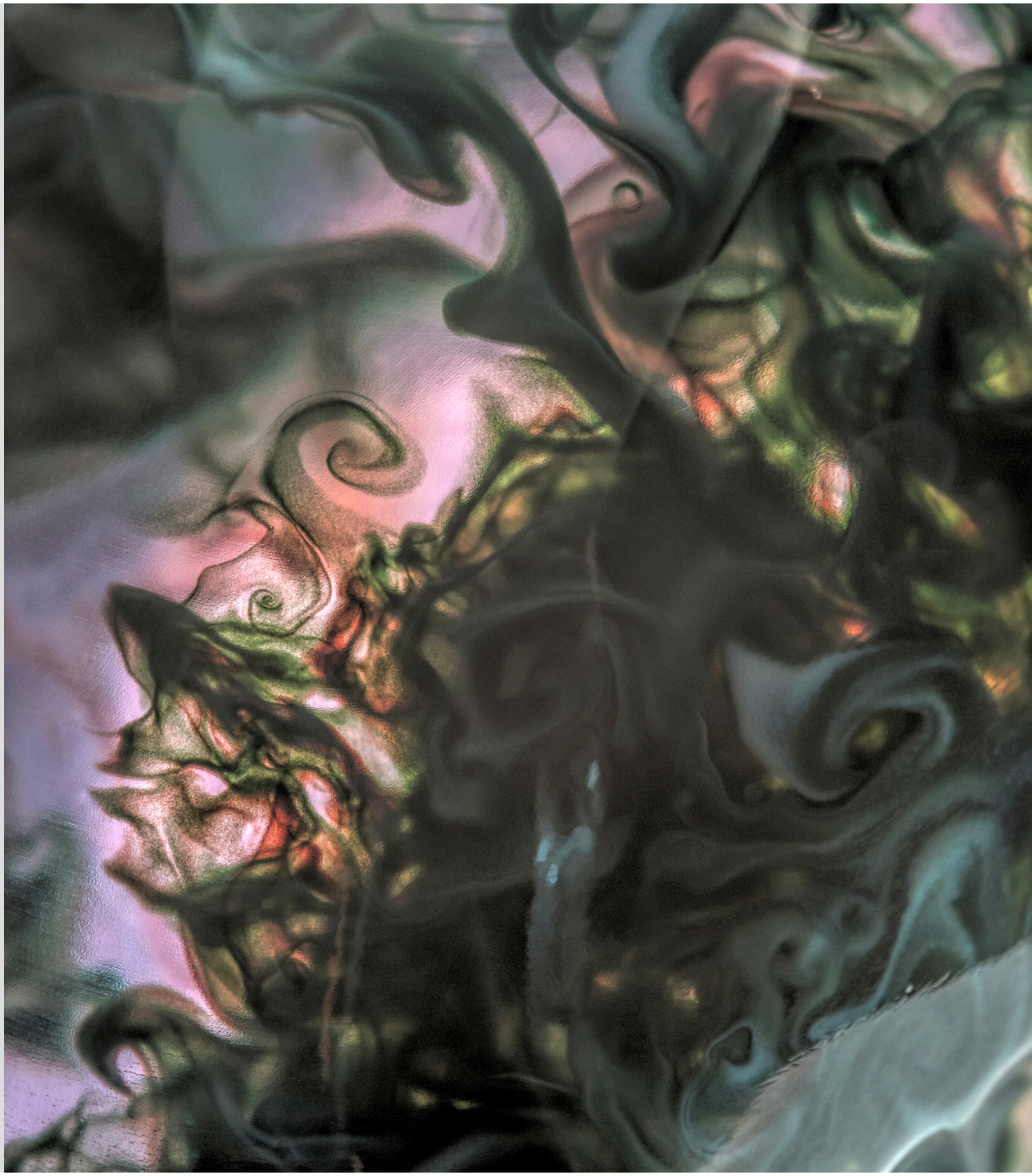
Mediated Matter

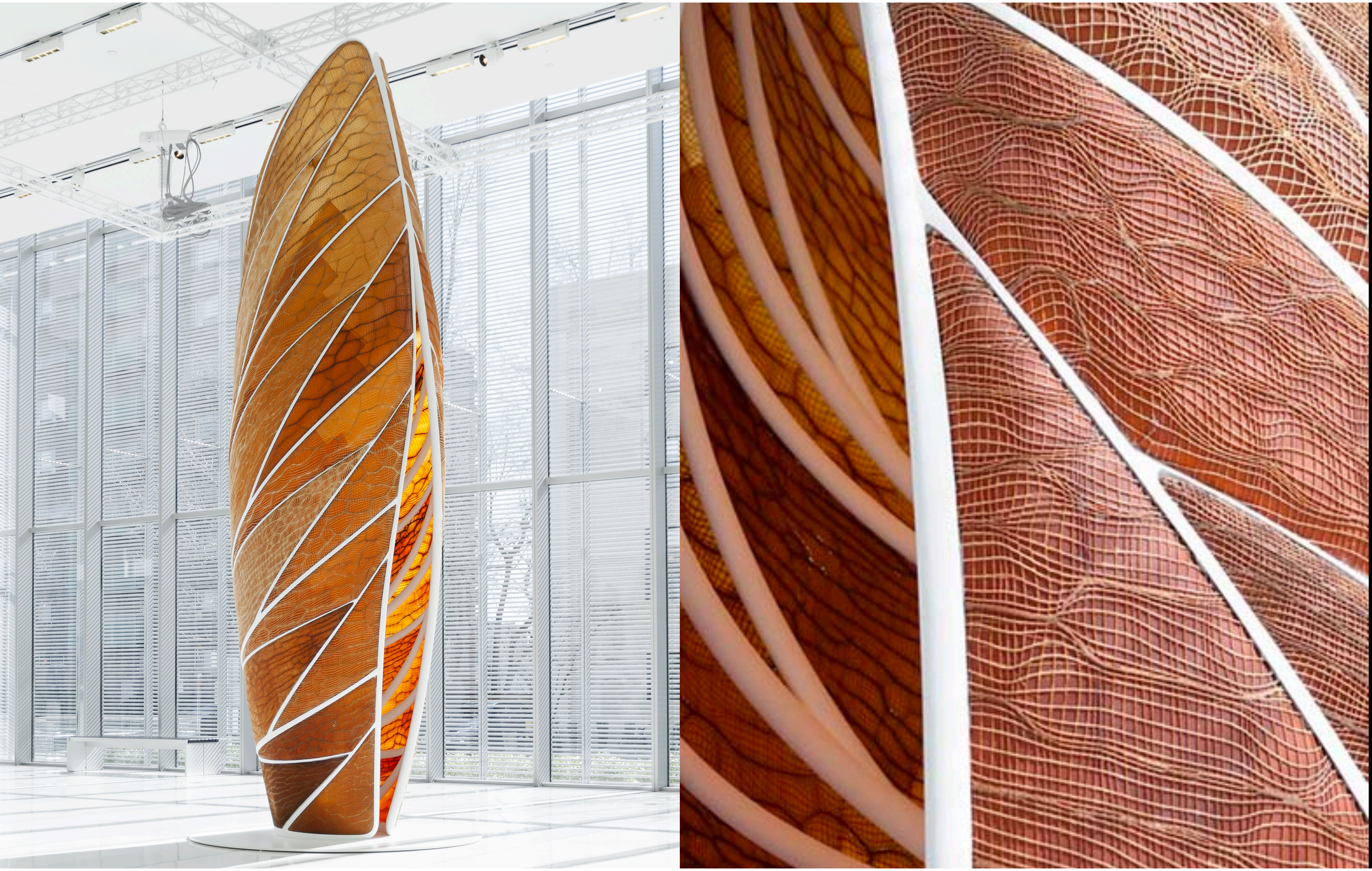


Neri Oxman



Neri Oxman

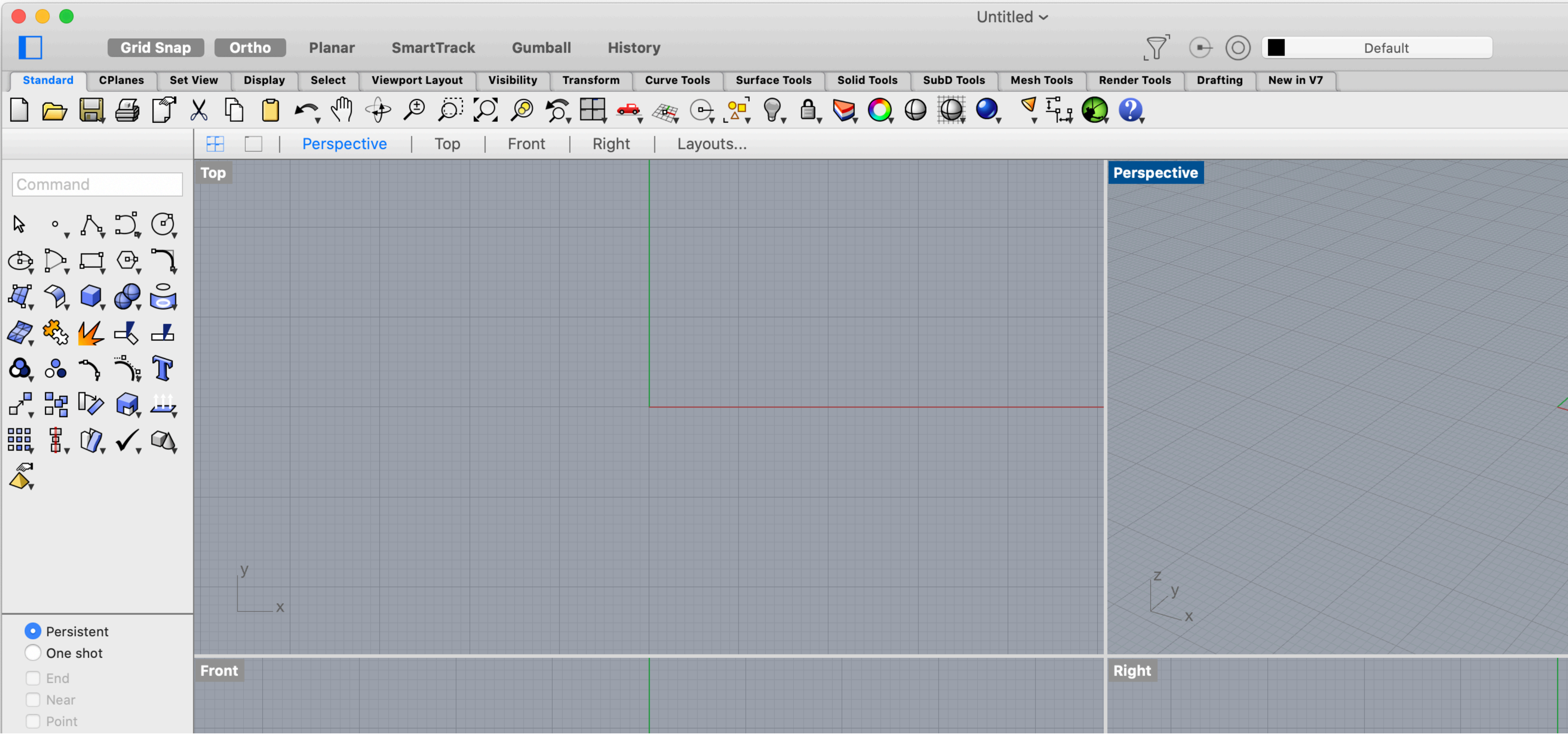




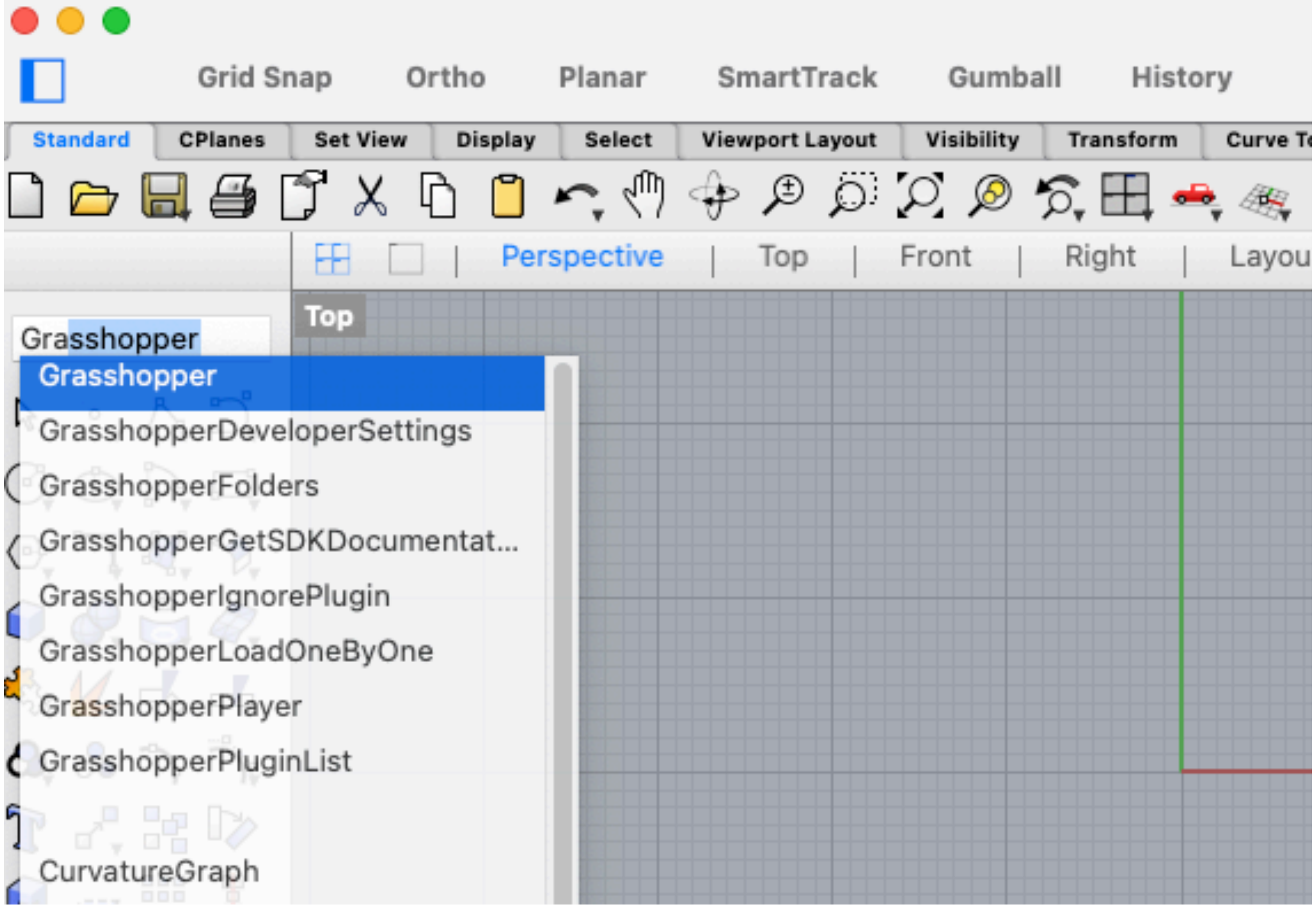
Neri Oxman

Rhino, Grasshopper, and Python

open up Rhino

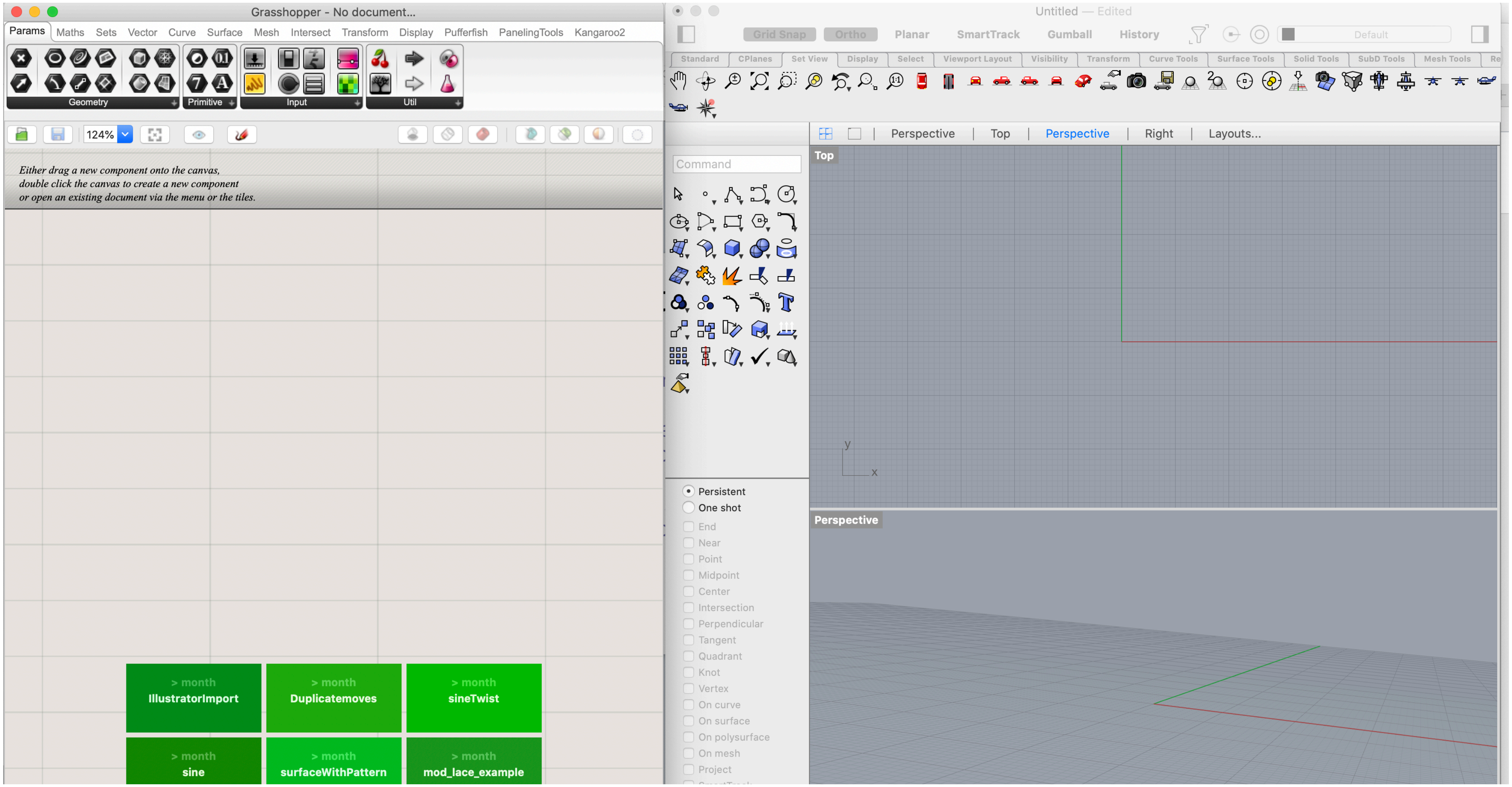


Open Grasshopper



Grasshopper

Set up so you can see both applications



Either drag a new component onto the canvas, double click the canvas to create a new component or open an existing document via the menu or the tiles.

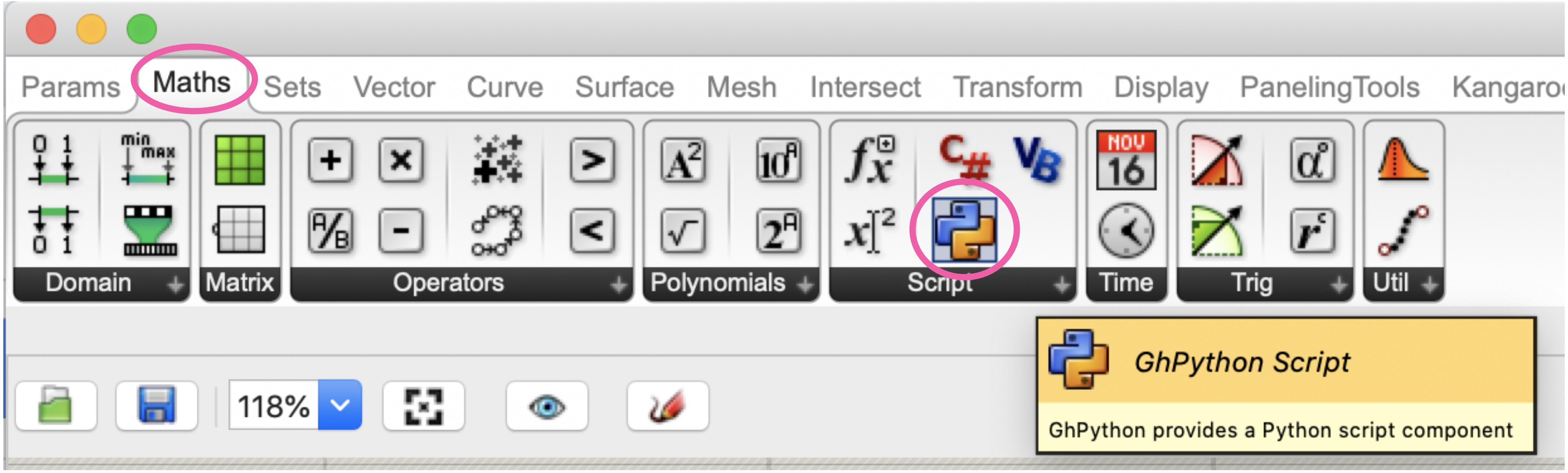
RhinoScript for Python Documentation

Open and bookmark:

https://developer.rhino3d.com/api/RhinoScriptSyntax

Points, Lines, and Surfaces

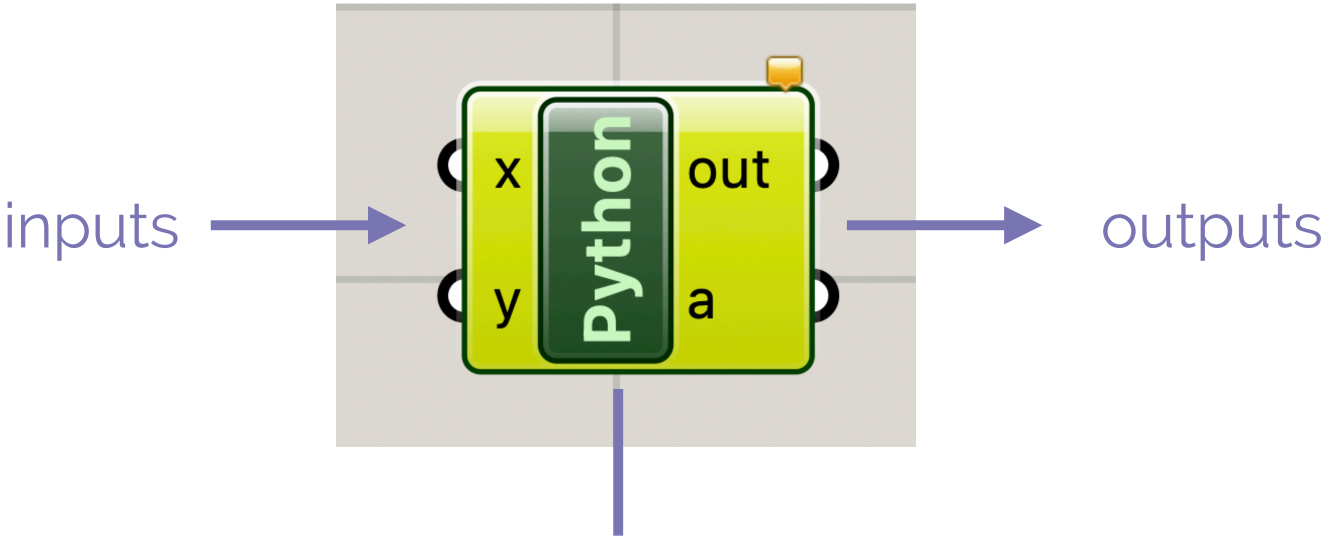
Python Code Block



GhPython Script

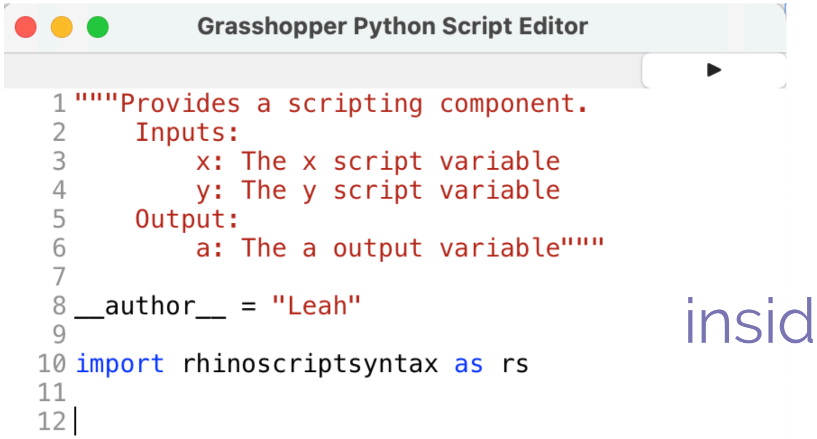
GhPython provides a Python script component

Python Code Block



inputs

outputs

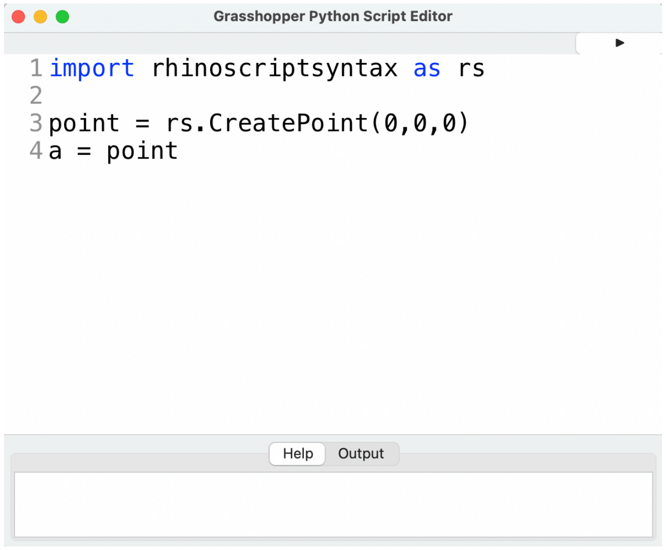


```
"""Provides a scripting component.
    Inputs:
        x: The x script variable
        y: The y script variable
    Output:
        a: The a output variable"""

__author__ = "Leah"

import rhinoscriptsyntax as rs

```

inside block: Python code

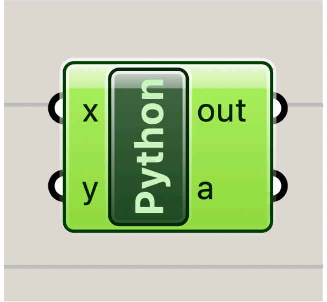
Points: rs.CreatePoint (x,y,z)

Python

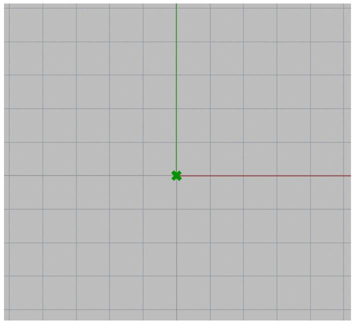


```
import rhinoscriptsyntax as rs

point = rs.CreatePoint(0,0,0)
a = point
```

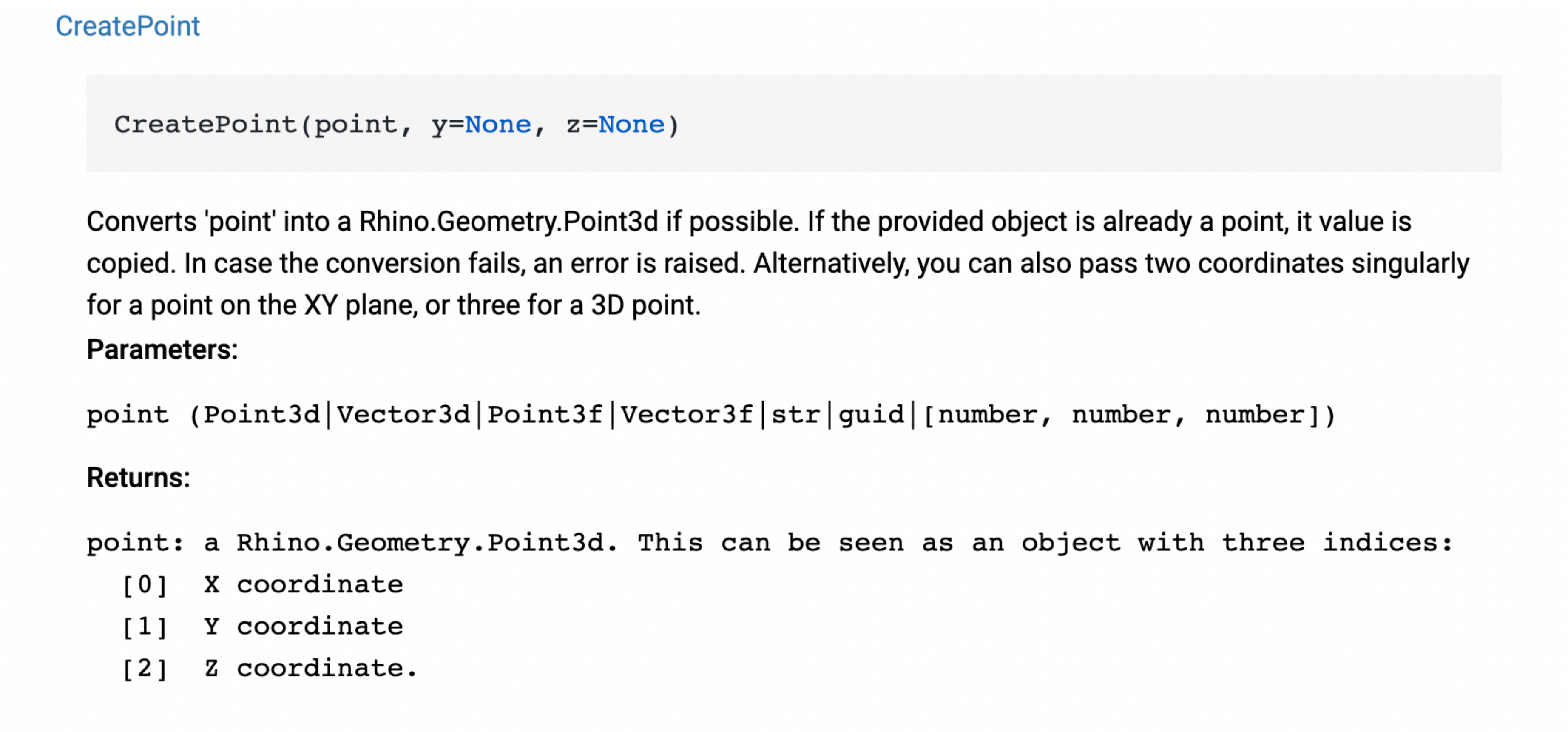
Grasshopper



Rhino



CreatePoint Documentation



CreatePoint

```
CreatePoint(point, y=None, z=None)
```

Converts 'point' into a Rhino.Geometry.Point3d if possible. If the provided object is already a point, it value is copied. In case the conversion fails, an error is raised. Alternatively, you can also pass two coordinates singularly for a point on the XY plane, or three for a 3D point.

Parameters:

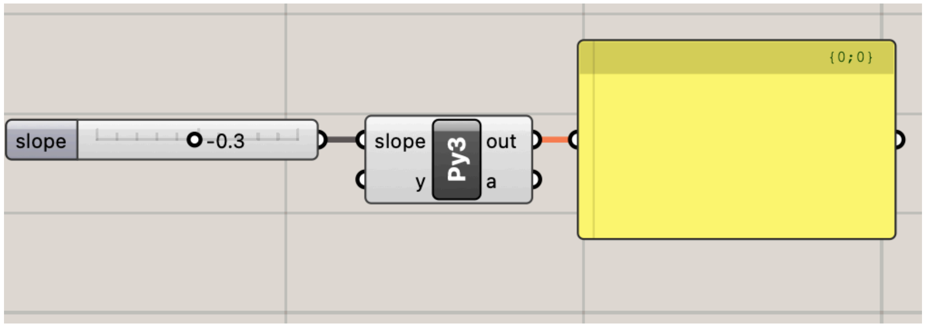
```
point (Point3d|Vector3d|Point3f|Vector3f|str|guid|[number, number, number])
```

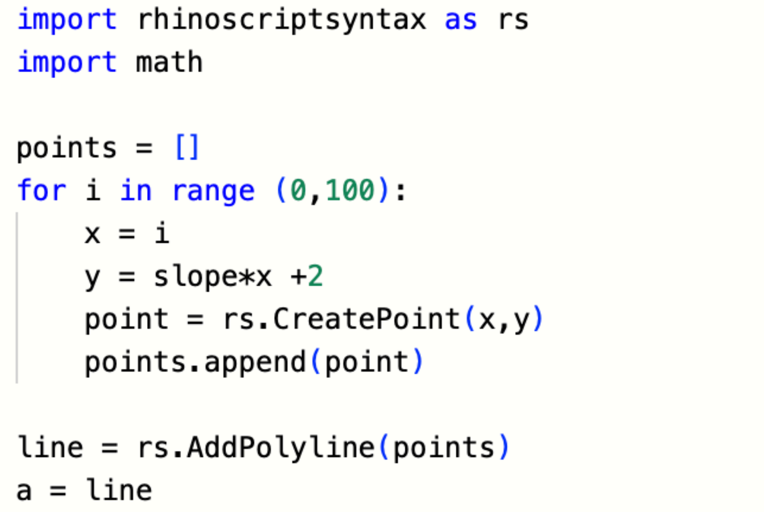
Returns:

```
point: a Rhino.Geometry.Point3d. This can be seen as an object with three indices:
  [0]  X coordinate
  [1]  Y coordinate
  [2]  Z coordinate.
```

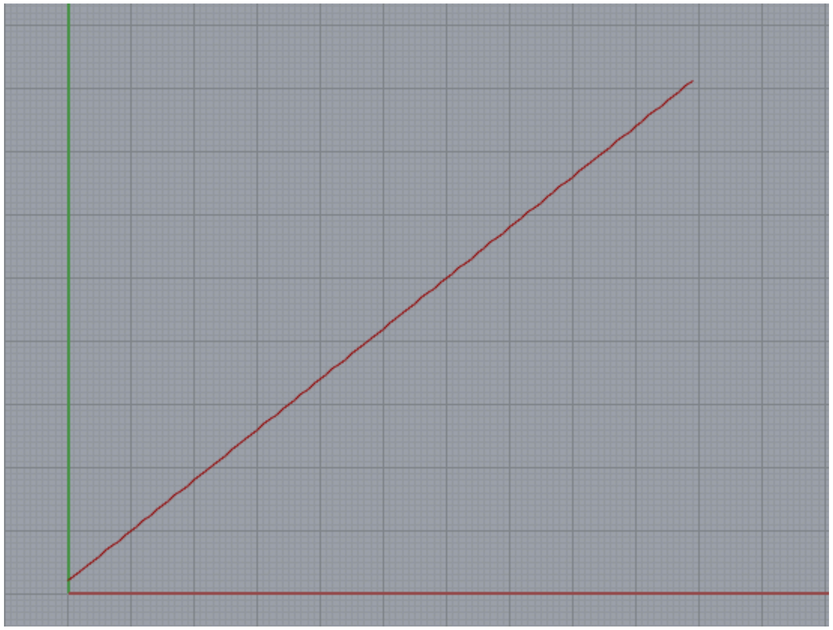
https://developer.rhino3d.com/api/RhinoScriptSyntax/#utility-CreatePoint

Points & AddPolyline

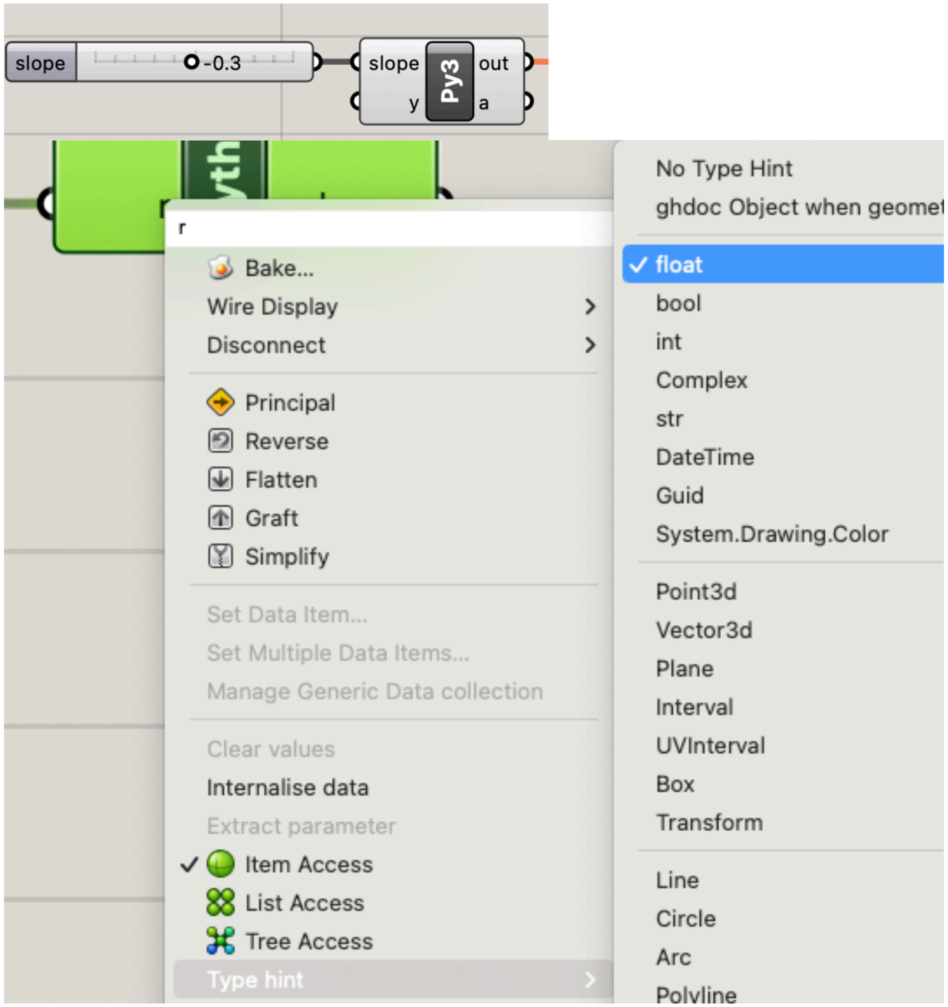




```
import rhinoscriptsyntax as rs
import math

points = []
for i in range (0,100):
    x = i
    y = slope*x +2
    point = rs.CreatePoint(x,y)
    points.append(point)

line = rs.AddPolyline(points)
a = line
```



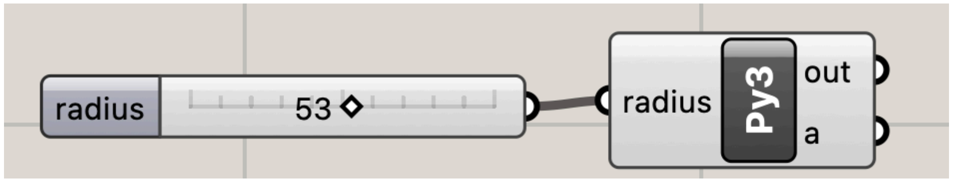
When adding the slope input

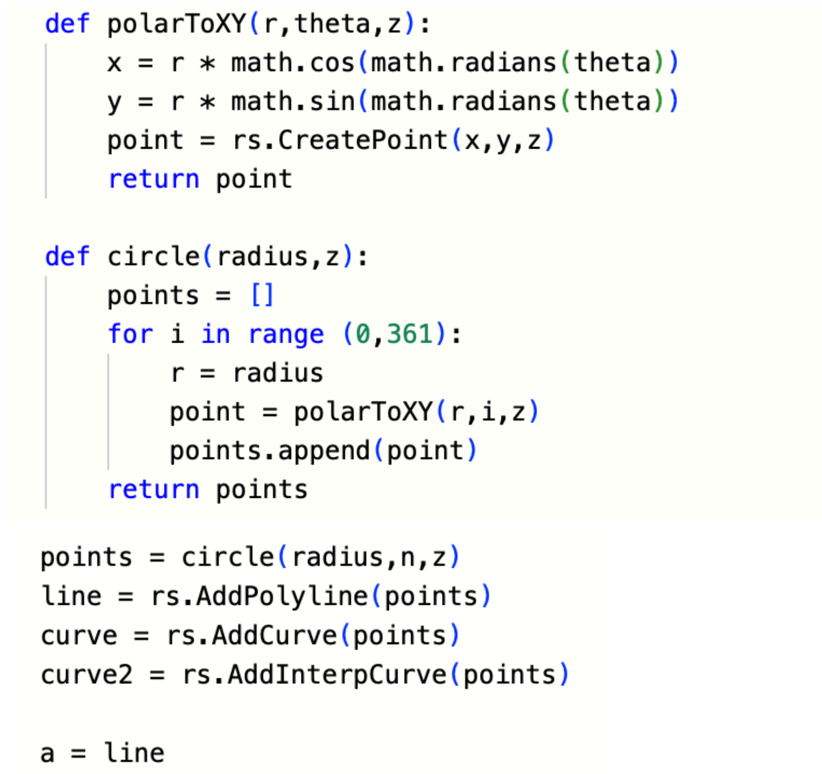


right click on variable
to rename variable &
access other options

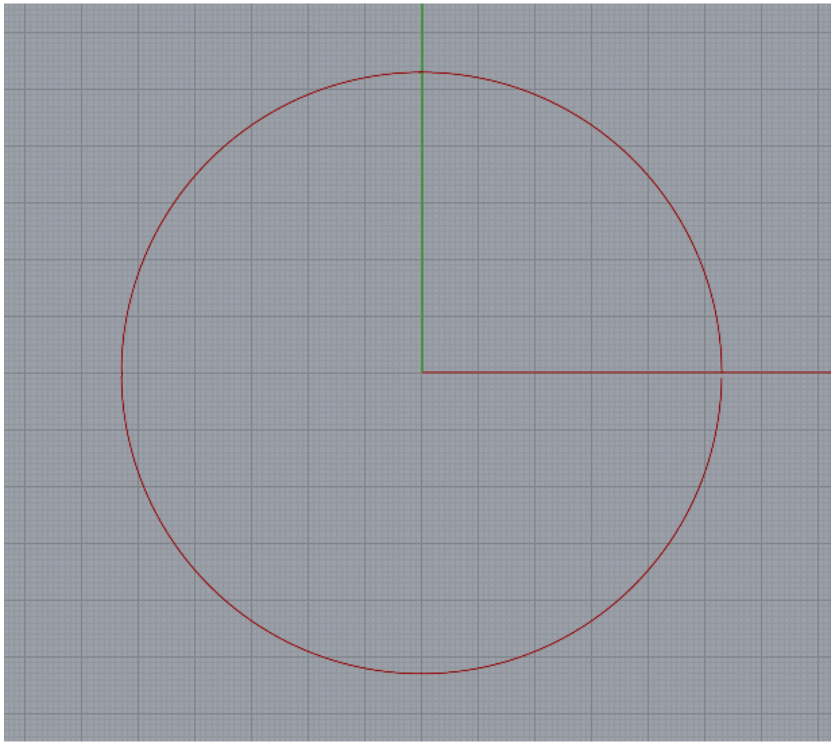
remember to give a **Type hint**
select Type hint
select float

Cylindrical coordinates & a circle

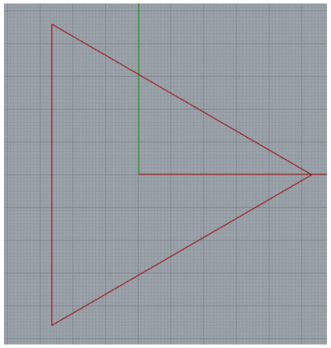


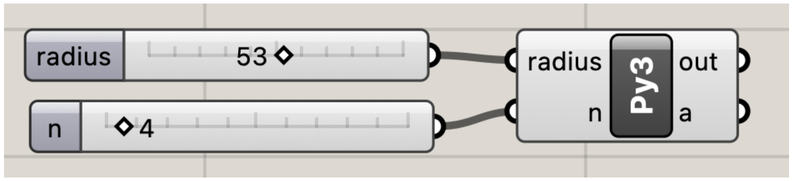


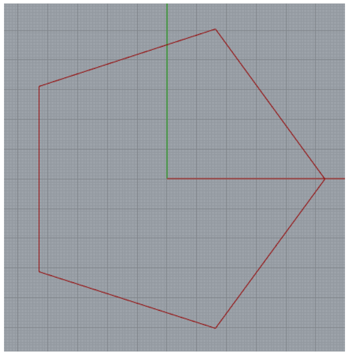
```
def polarToXY(r,theta,z):
    x = r * math.cos(math.radians(theta))
    y = r * math.sin(math.radians(theta))
    point = rs.CreatePoint(x,y,z)
    return point

def circle(radius,z):
    points = []
    for i in range (0,361):
        r = radius
        point = polarToXY(r,i,z)
        points.append(point)
    return points

points = circle(radius,n,z)
line = rs.AddPolyline(points)
curve = rs.AddCurve(points)
curve2 = rs.AddInterpCurve(points)

a = line
```

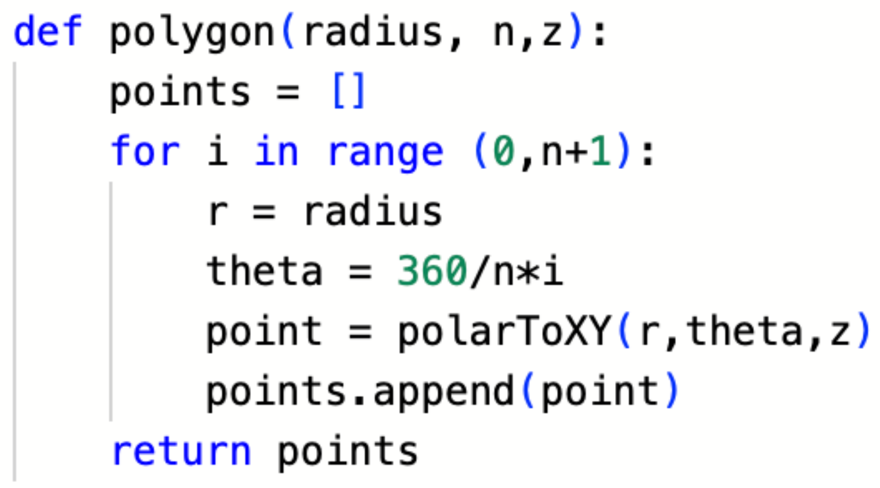


Polygons

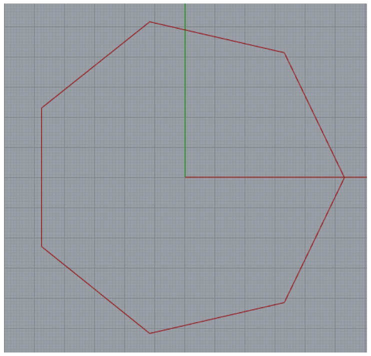






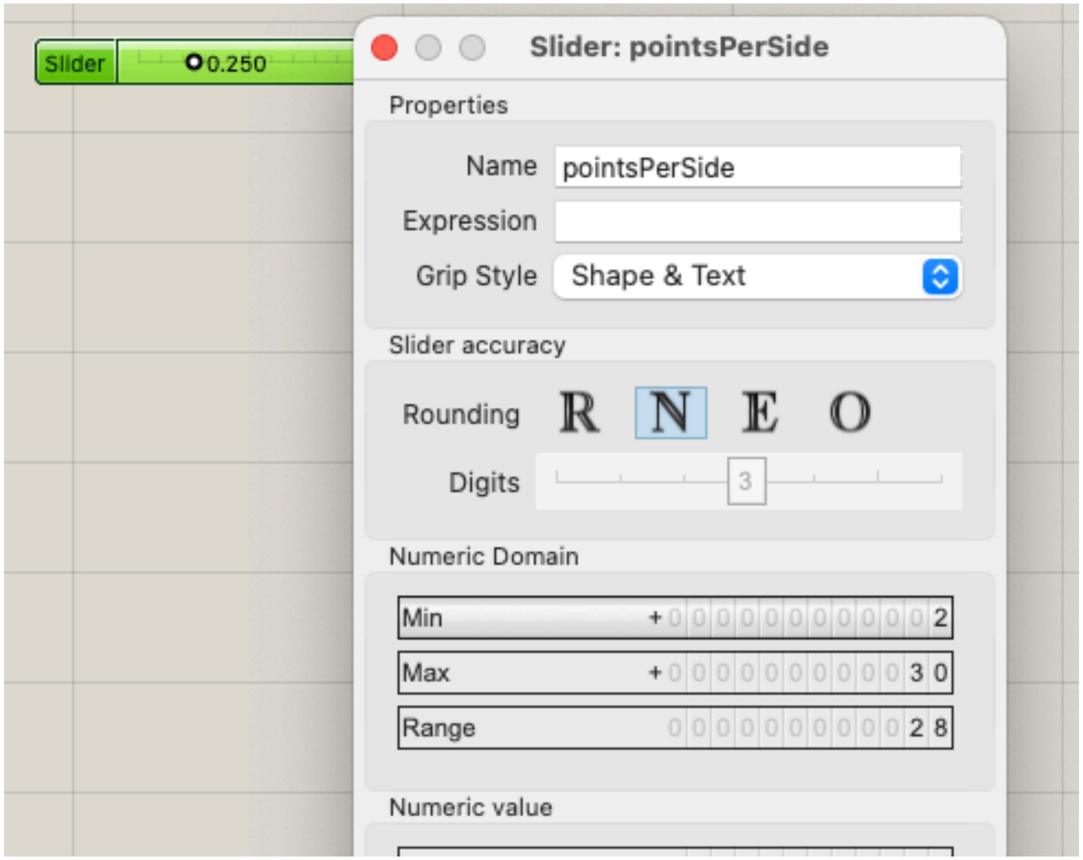


```
def polygon(radius, n,z):
    points = []
    for i in range (0,n+1):
        r = radius
        theta = 360/n*i
        point = polarToXY(r,theta,z)
        points.append(point)
    return points
```



Edit the n number slider to restrict range of values: 3 to 20

double click on slider for options



Slider: pointsPerSide

Properties

Name: pointsPerSide

Expression:

Grip Style: Shape & Text

Slider accuracy

Rounding: R [✓ N] E O

Digits: 3

Numeric Domain

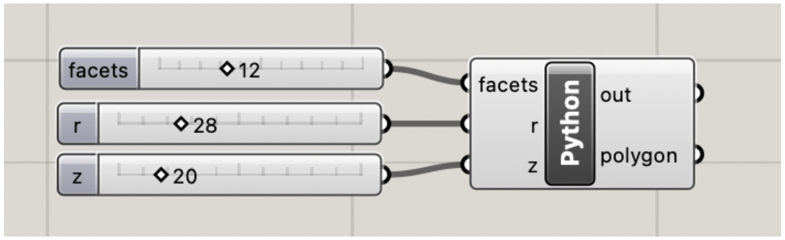
Min: +000000000002

Max: +000000000030

Range: 000000000028

Numeric value

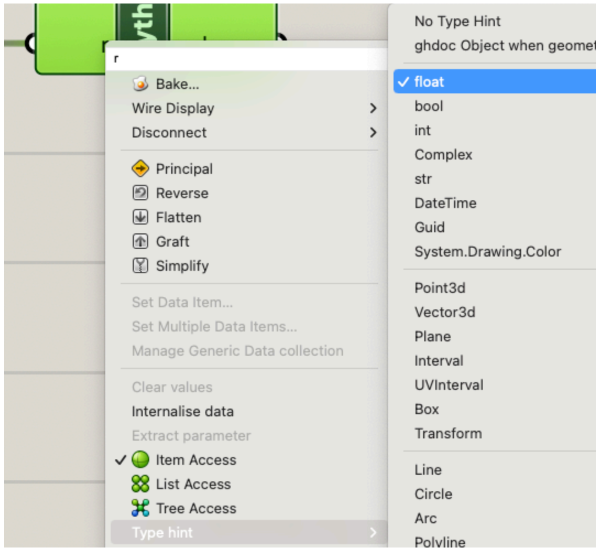
Add a z input slider



facets ◇12

r ◇28

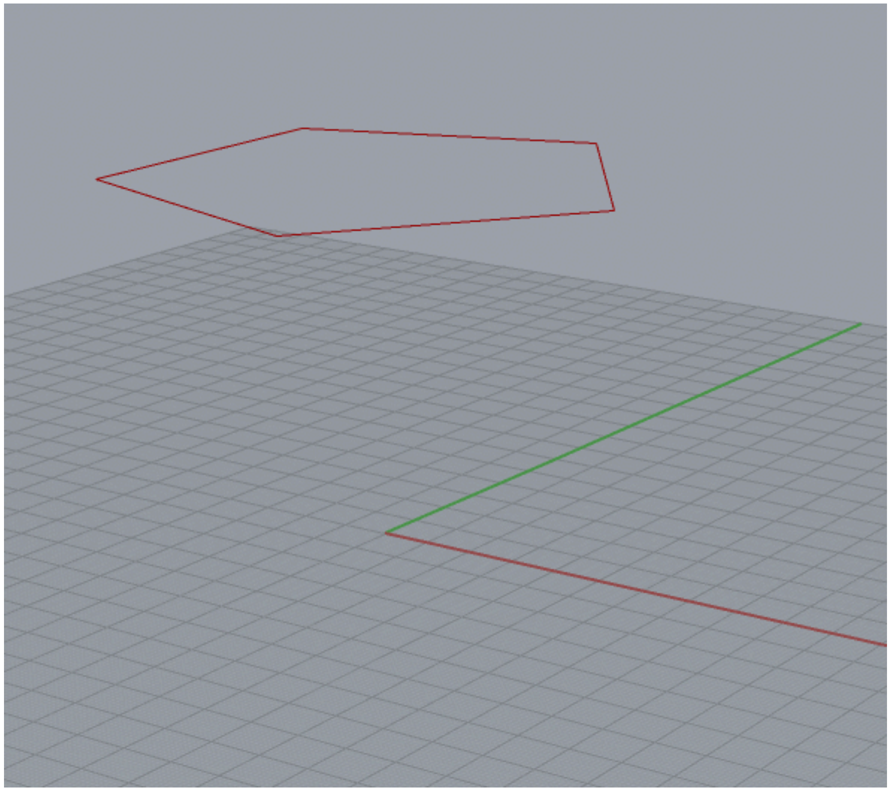
z ◇20



✓ Item Access

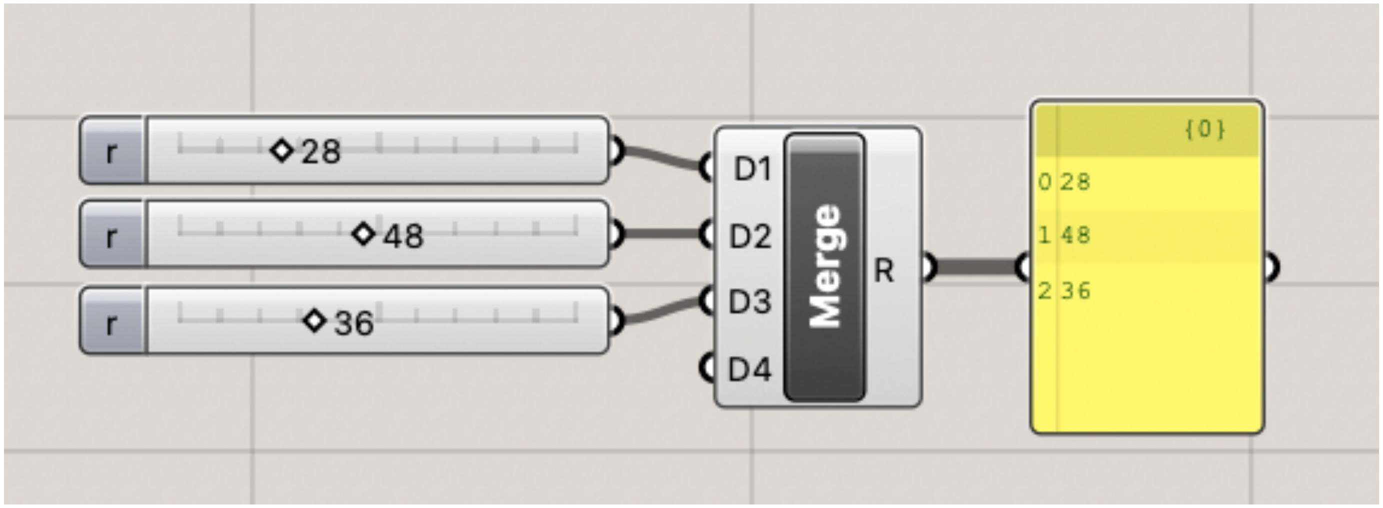
✓ float

remember Type hint
select **float**



Now we’re going to create several polygons with different rs, ns, and zs

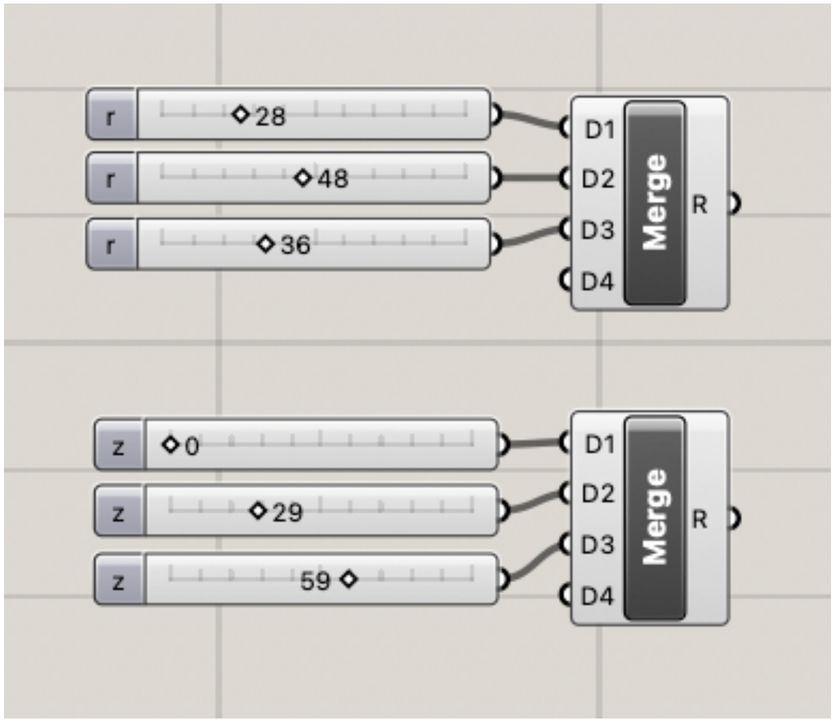
Lists in Grasshopper using Merge

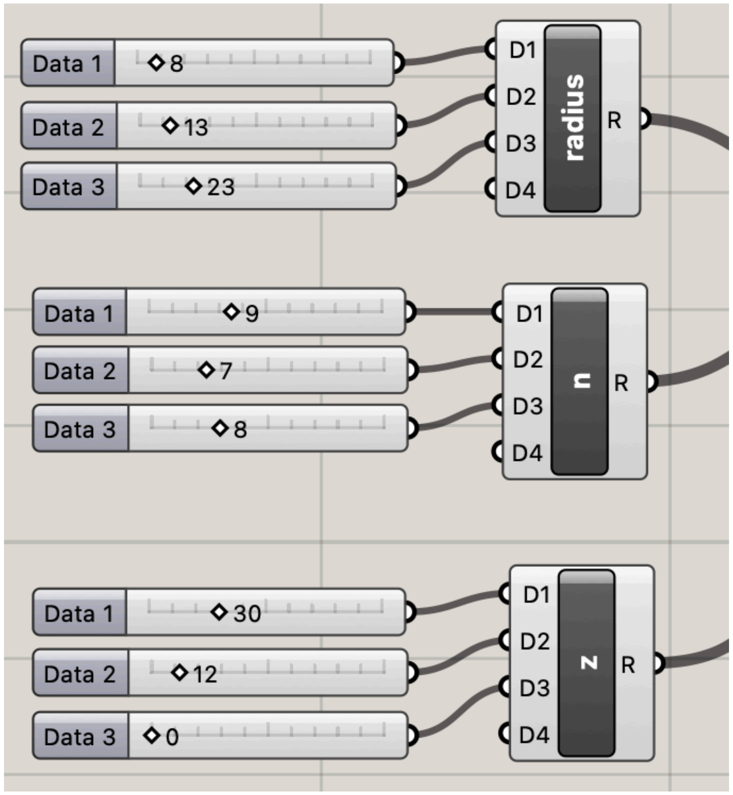


The Merge block is a good way to create lists with specific entries.

https://grasshopperdocs.com/components/grasshoppersets/merge.html

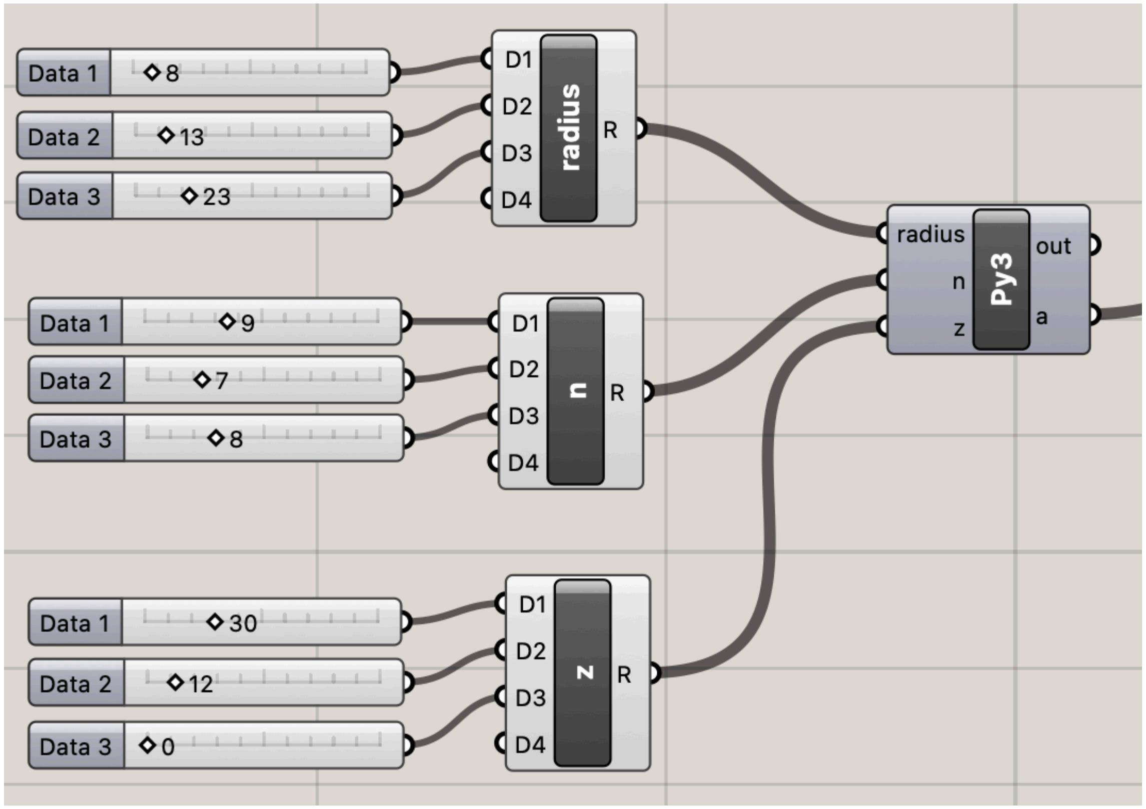
Create Merge blocks for r, n, and z with 3 entries each

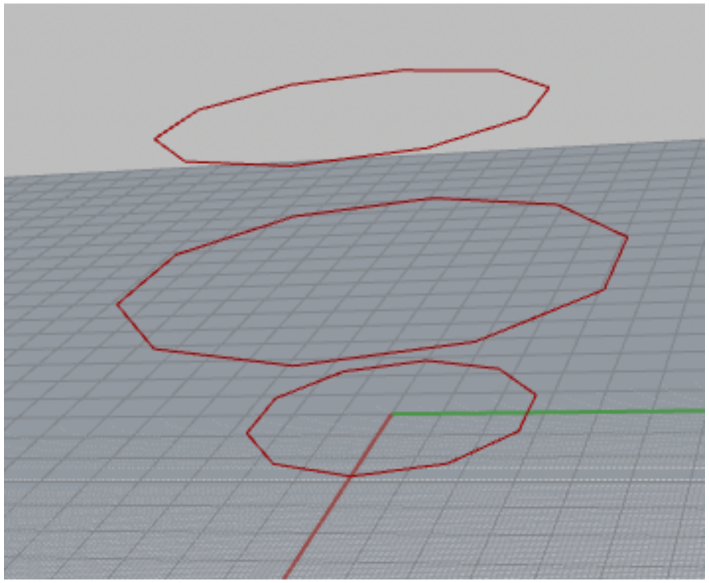




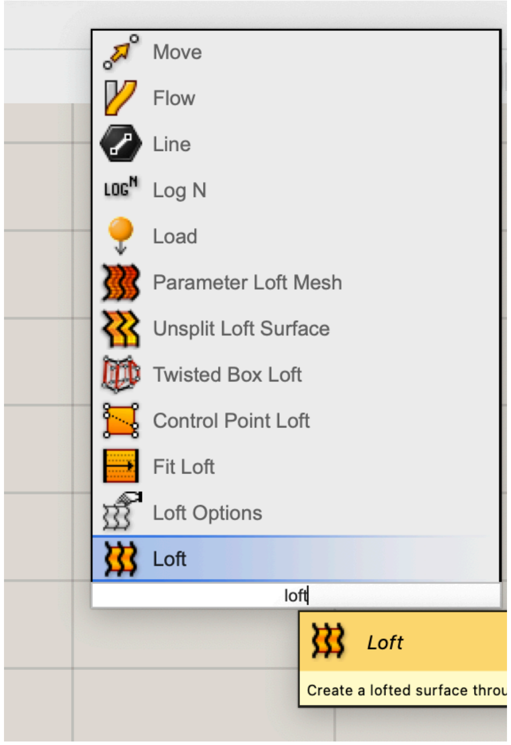
rename them

Connect to Python block

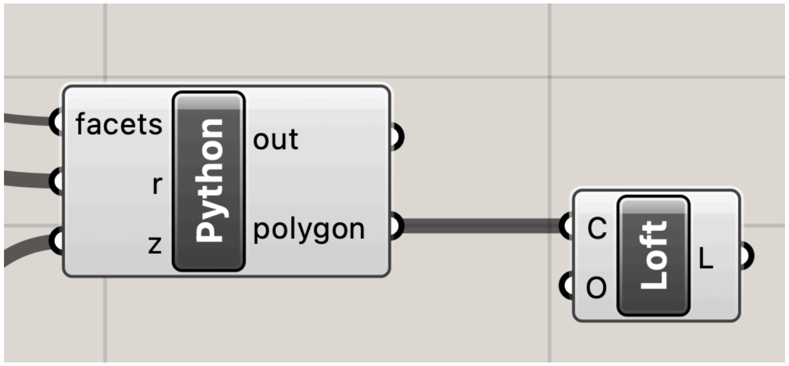


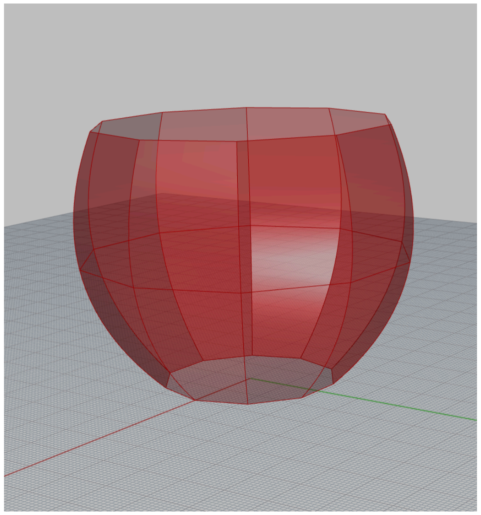


Create a surface using Loft

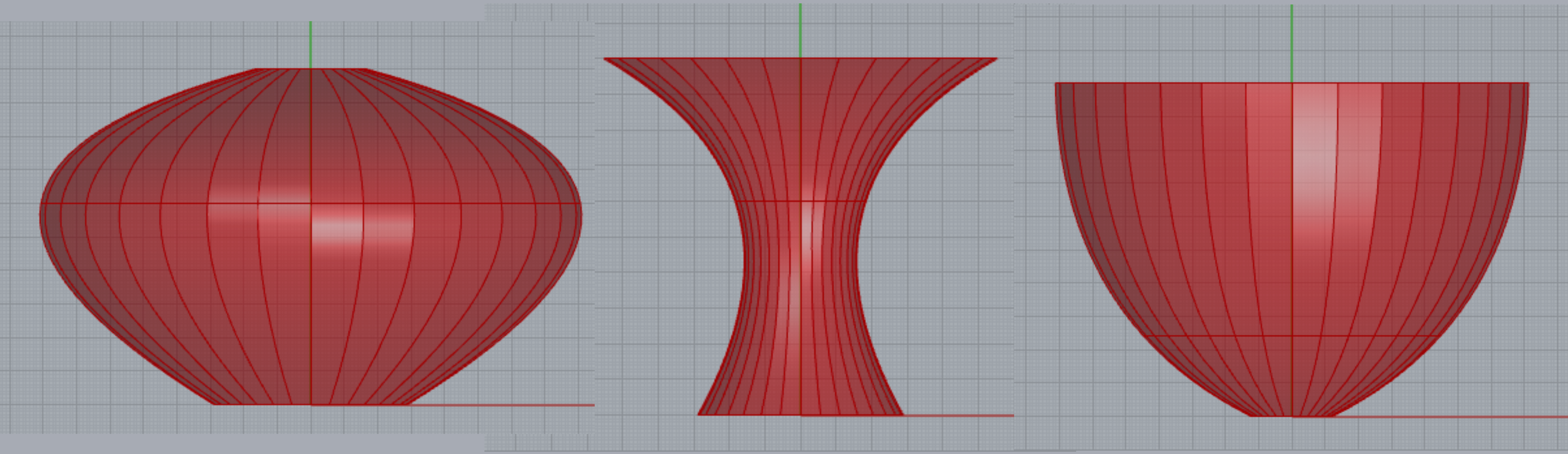


Create a lofted surface throu

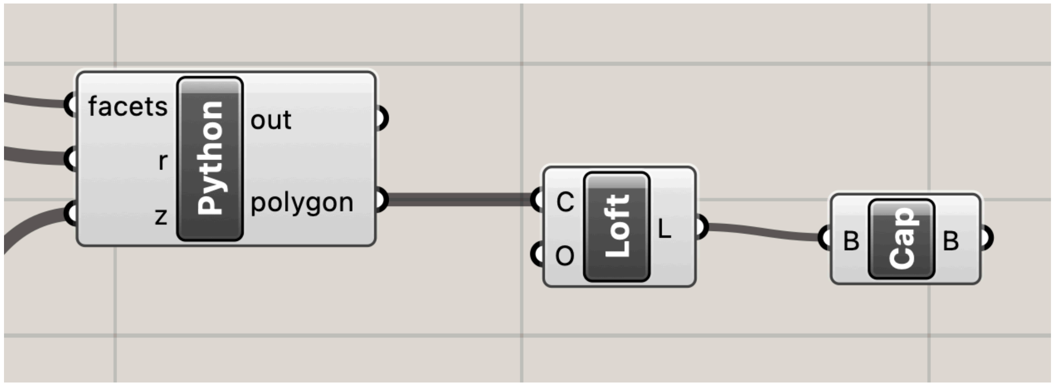


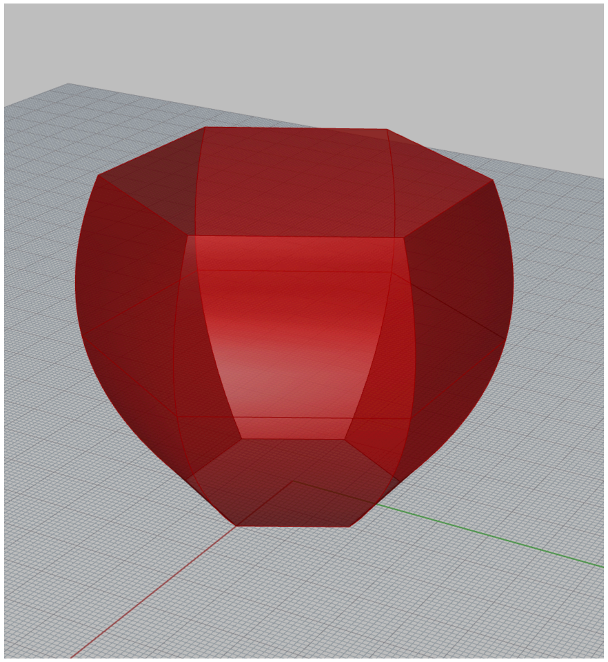


Play with sliders

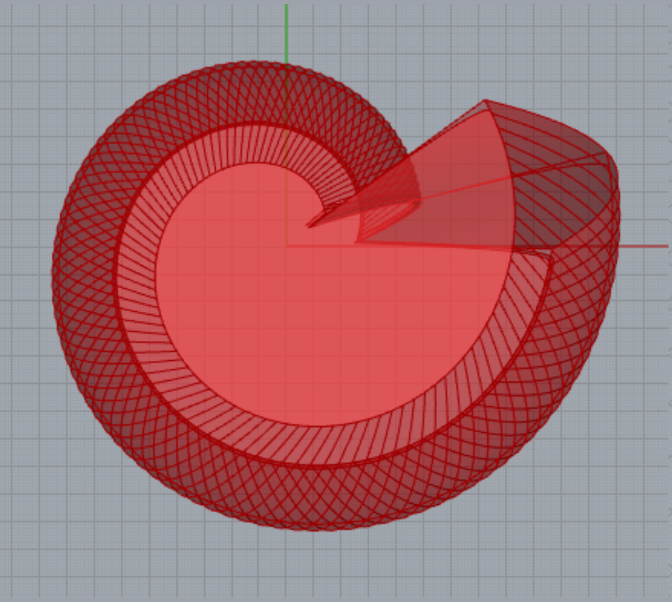


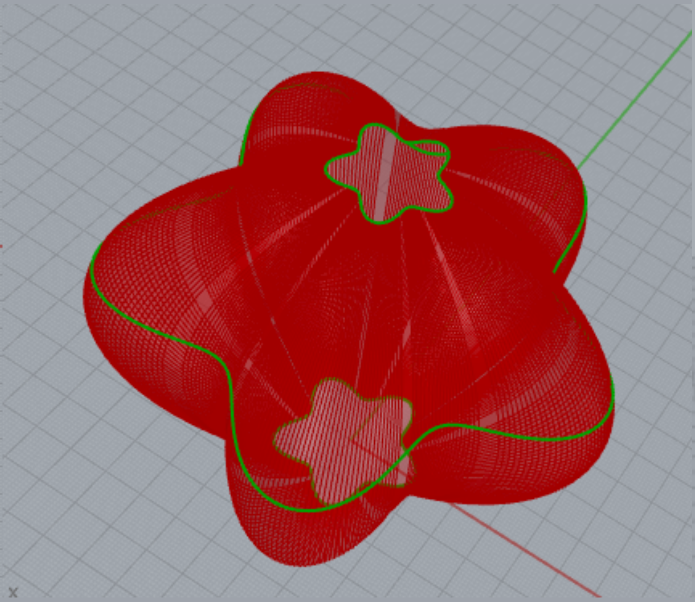
Create a solid with Cap

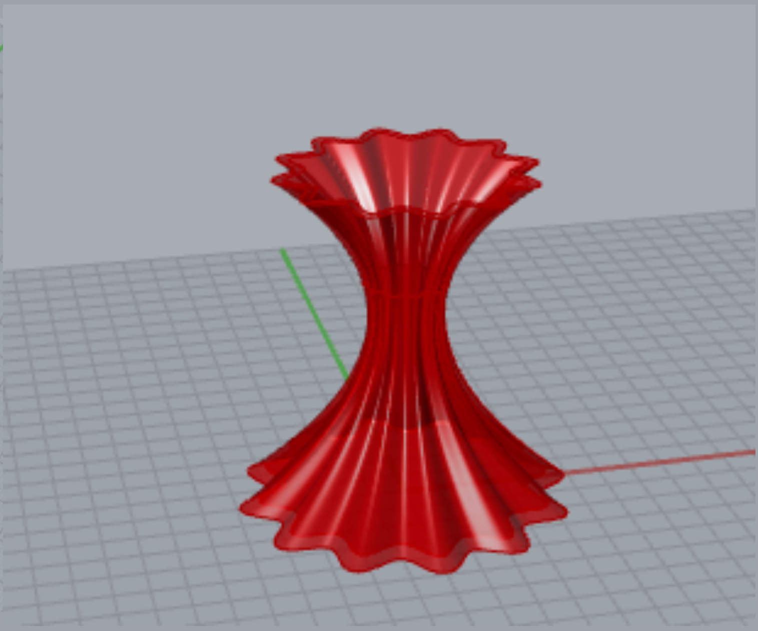




more interesting surfaces: r = f(angle)







Thank you!

CS 491 and 591
Professor: Leah Buechley
https://handandmachine.org/classes/computational_fabrication/