

# Computational Fabrication

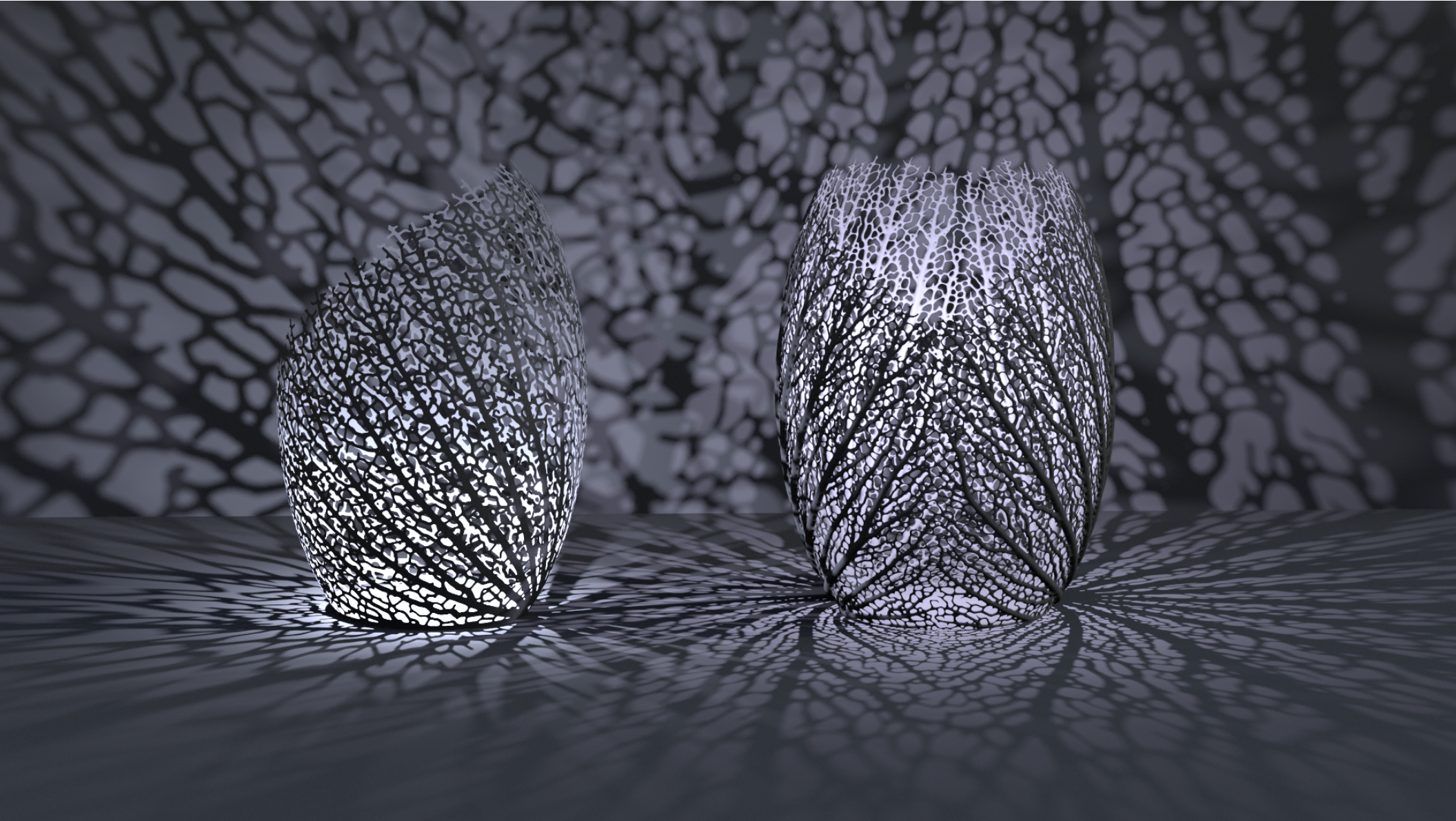
CS 491 and 591

Professor: Leah Buechley

[https://handandmachine.cs.unm.edu/classes/Computational\\_Fabrication\\_Spring2021/](https://handandmachine.cs.unm.edu/classes/Computational_Fabrication_Spring2021/)

# Weekly Designer: Nervous System

<https://n-e-r-v-o-u-s.com/index.php>

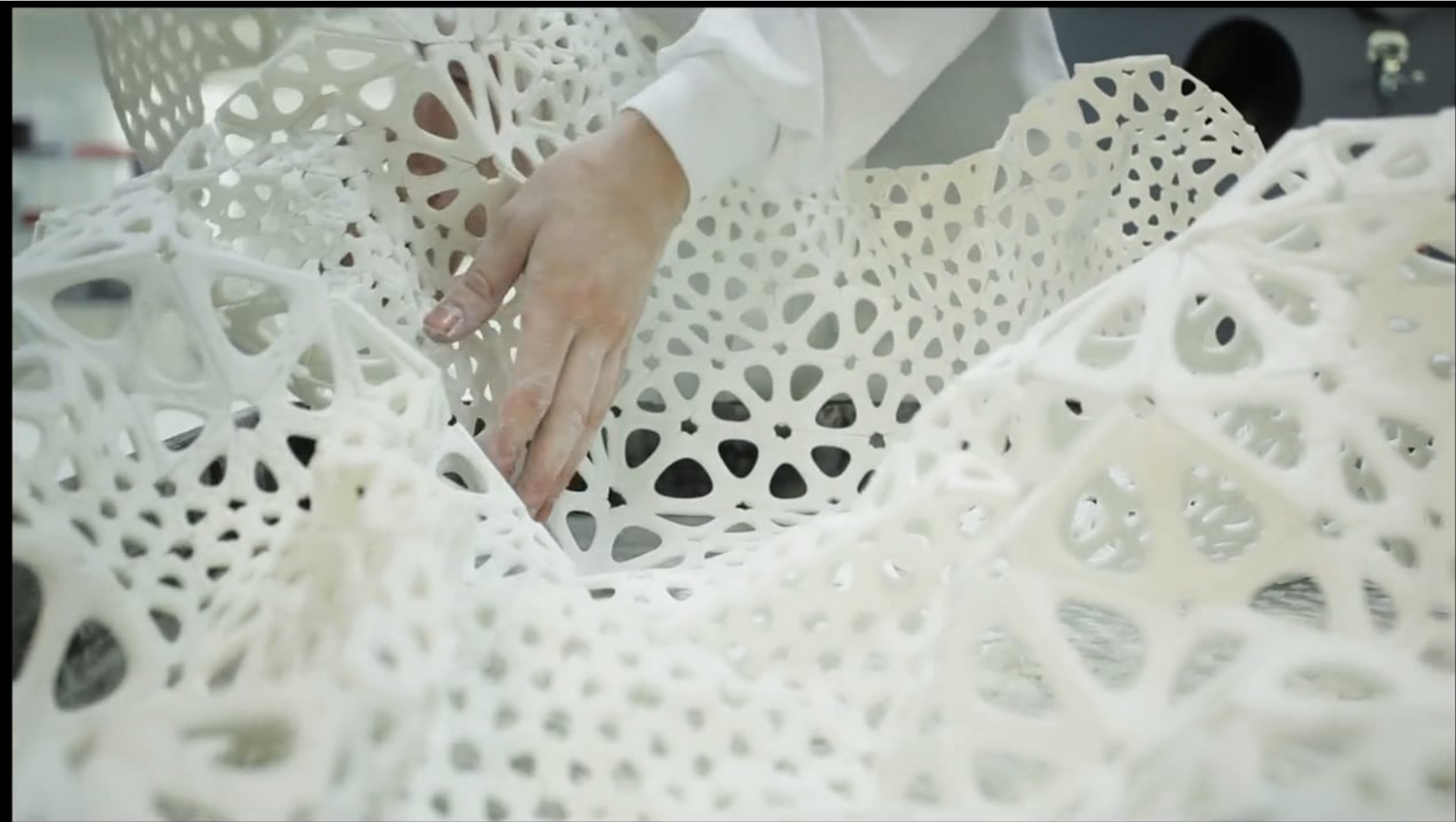






Nervous System









New blue pigments  
to dye for *p. 424*

Snow's "two cultures"  
at 60 *p. 430*

Hybridization helps killfish  
resist toxins *pp. 433 & 455*

# Science

\$15  
3 MAY 2019  
sciencemag.org

AAAS

## ENGINEERED VASCULATURE

Microfabricated  
paths for blood  
and oxygen  
flow *p. 458*



<https://n-e-r-v-o-u-s.com/blog/?p=8433>

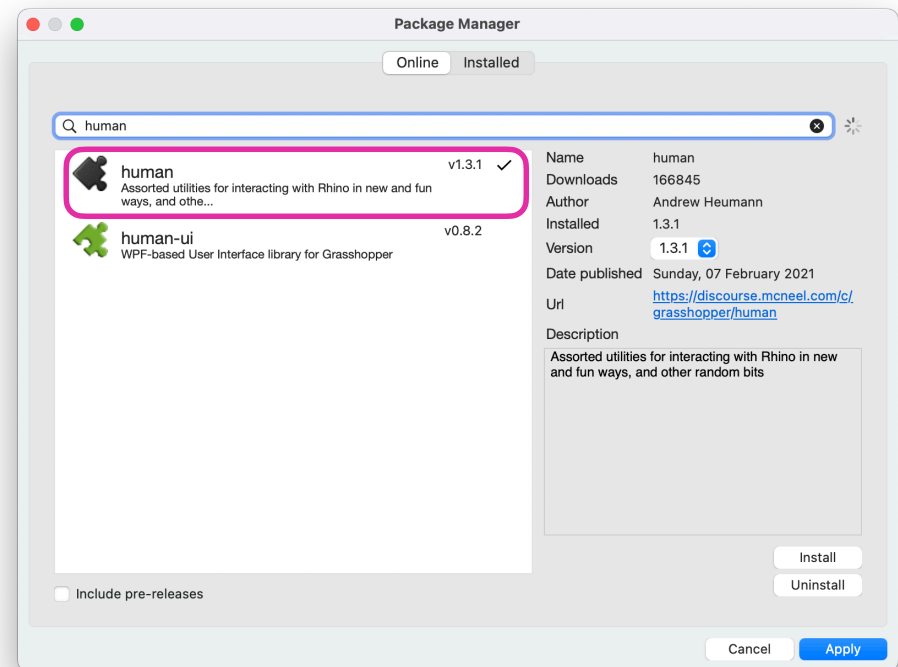


**Data Driven Design cont.**

What I messed up in class:  
installing the **human** GH plugin

# Install the human GH plugin

- **IMPORTANT:** must follow these exact steps to install the correct version of the library. Do not install the Food4Rhino version.
- In Rhino, open the Package Manager by typing "Package Manager" in the command line.
- Choose "human" v1.3.1 in the box that pops up and click Apply.
- Quit and restart Rhino.

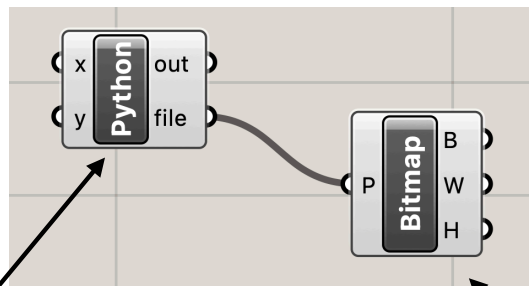




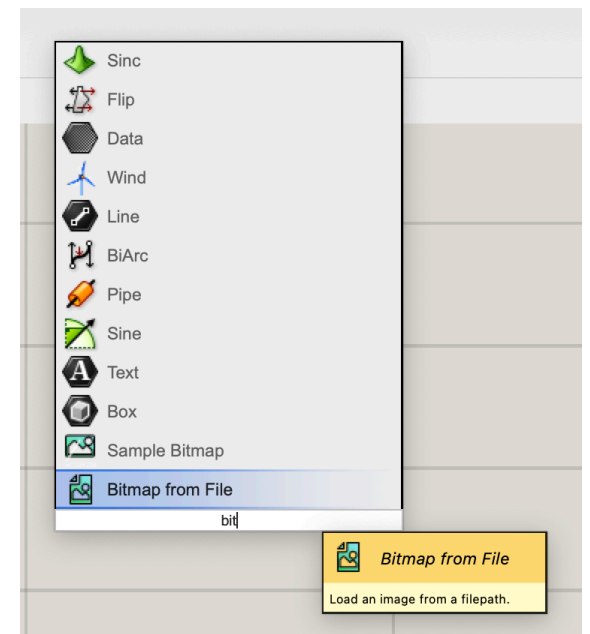
# Open the bitmap in GH

specify path

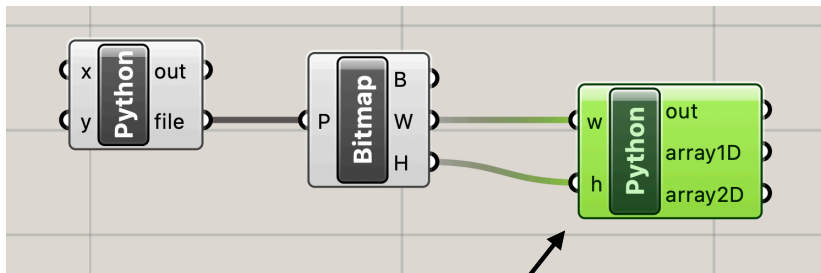
```
1 import rhinoscriptsyntax as rs
2
3 filter = "IMG file (*.bmp)|*.pbm|*.png|All Files (*.*)|*.*|*"
4 file = rs.OpenFileName("Open Image File", filter)
```



open bitmap block  
(requires human plugin)



# Create point arrays

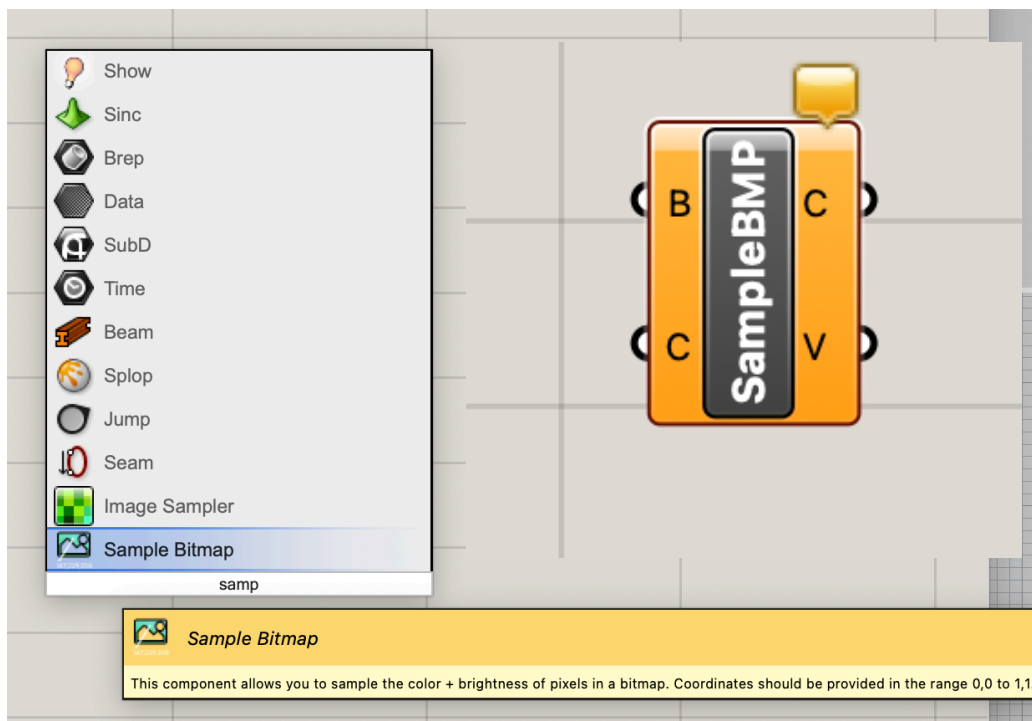


The 1D array creates a list of points in the range 0-1. This will be used to access pixel information.

```
1 import rhinoscriptsyntax as rs
2
3 array2D = []
4 array1D = []
5 for i in range(0,w,10):
6     row = []
7     for j in range(0,h,10):
8         row.append(rs.CreatePoint(i,j,0))
9         array1D.append(rs.CreatePoint(i/float(w),j/float(h),0))
10    array2D.append(row)
```

The 2D array creates an array of points in that correspond to the size of the bitmap. This will be used to create the final geometry.

# Get pixel information using Sample Bitmap



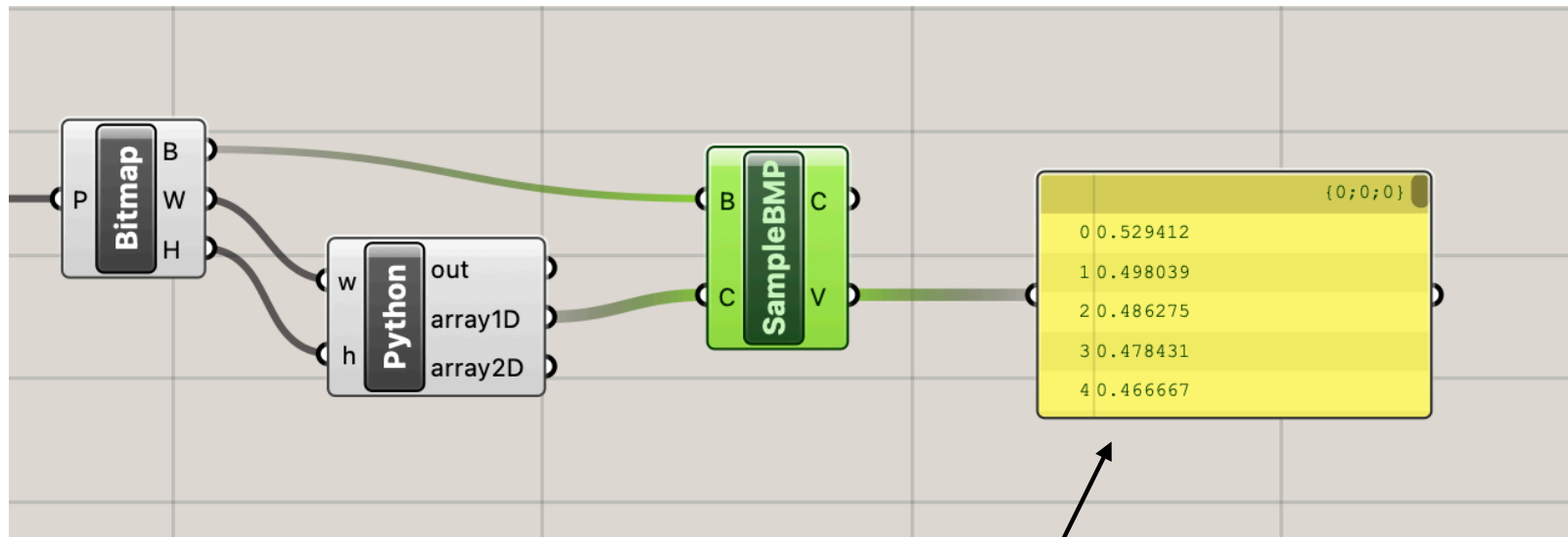
returns pixel color (C) and  
brightness (V) information

B input = bitmap

C input = point/pixel index  
(in range of 0-1)



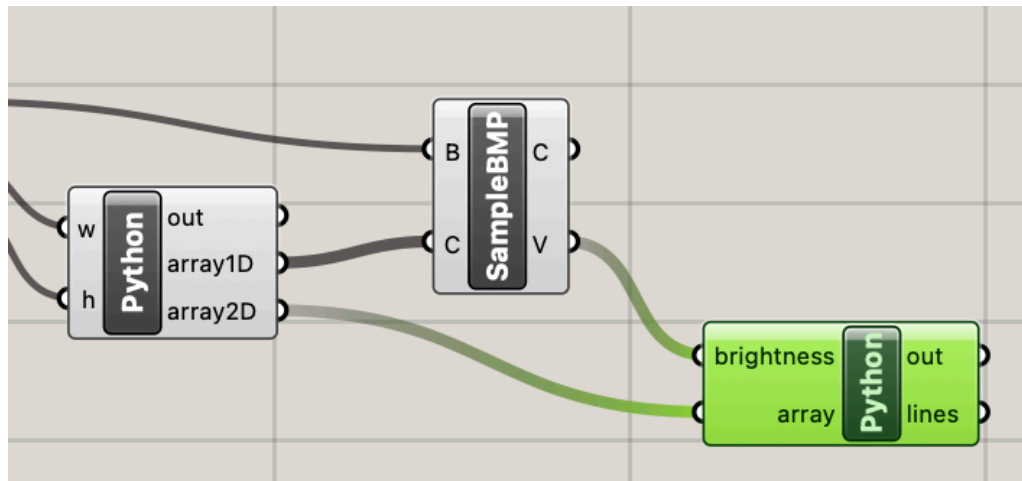
# Get pixel information



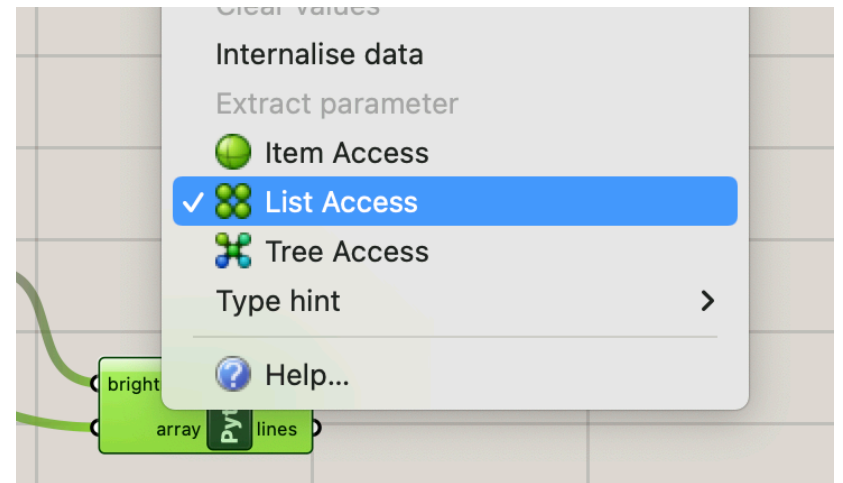
brightness of each pixel  
note: just using a yellow Panel  
text box to look at the data.

Generate geometry from pixel info

# Important: List Access vs. Item Access



connect 2D array & brightness info to new python block



choose list access for both inputs  
also set correct Type hints  
brightness: float  
array: Point

# Brightness = z component of new point

```
1 import rhinoscriptsyntax as rs
2
3 lines = []
4
5 b = 0
6 for i in range (0,len(array)):
7     points = []
8     for j in range (0,len(array[i])):
9         point = rs.CreatePoint(array[i][j].X, array[i][j].Y, brightness[b]*500)
10        points.append(point)
11        b = b+1
12    line = rs.AddCurve(points)
13    lines.append(line)
```

add a multiplier to z component  
(here 500) to get a more  
dramatic effect

# Array of points for each image row

```
1 import rhinoscriptsyntax as rs
2
3 lines = []
4
5 b = 0
6 for i in range (0,len(array)):
7     points = []
8     for j in range (0,len(array[i])):
9         point = rs.CreatePoint(array[i][j].X, array[i][j].Y, brightness[b]*500)
10        points.append(point)
```

# Line for each image row

```
1 import rhinoscriptsyntax as rs
2
3 lines = []
4
5 b = 0
6 for i in range (0,len(array)):
7     points = []
8     for j in range (0,len(array[i])):
9         point = rs.CreatePoint(array[i][j].X, array[i][j].Y, brightness[b]*500)
10        points.append(point)
11        b = b+1
12    line = rs.AddCurve(points)
```

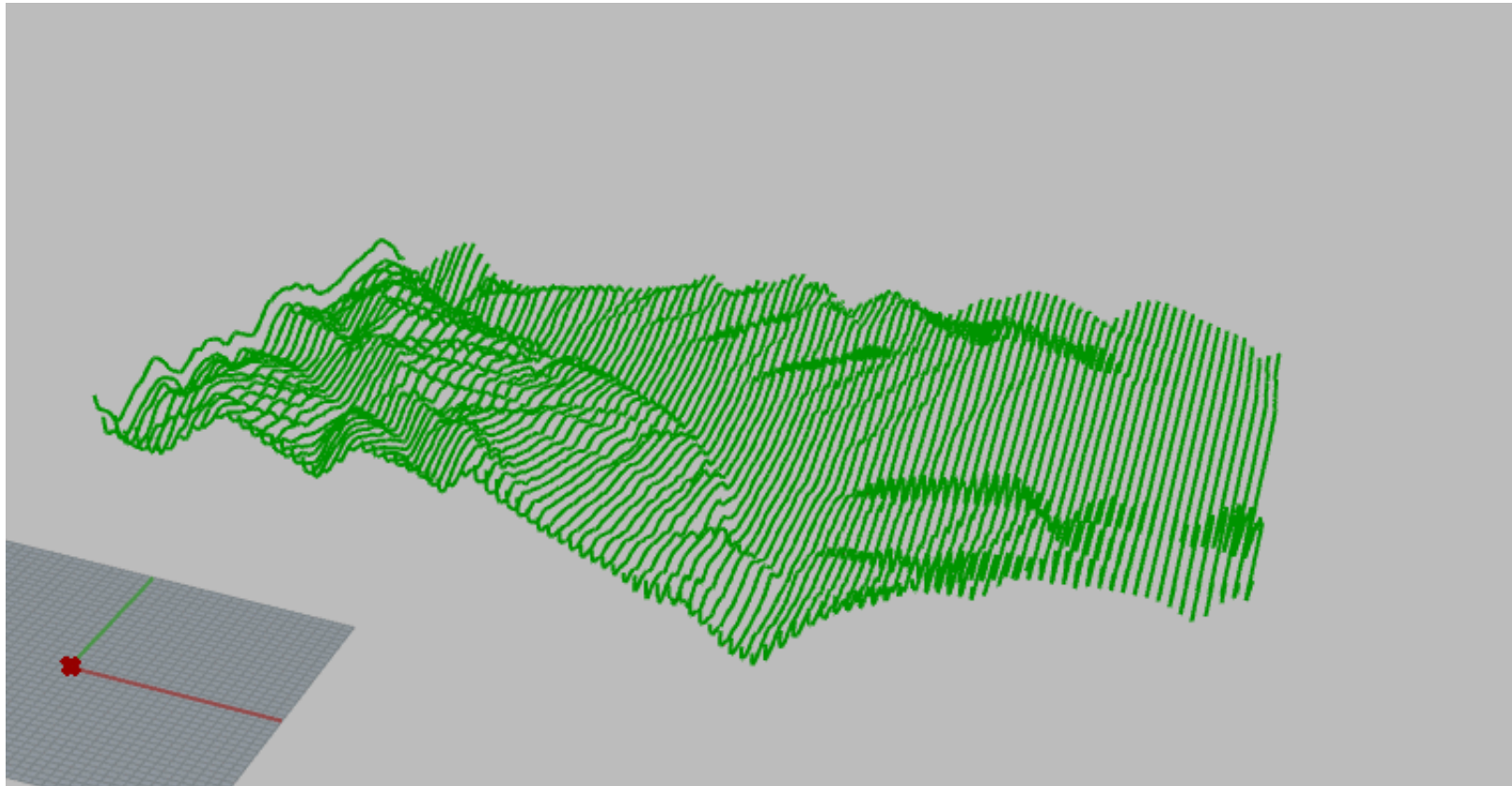
# Array of lines is final output

```
1 import rhinoscriptsyntax as rs
2
3 lines = []
4
5 b = 0
6 for i in range (0,len(array)):
7     points = []
8     for j in range (0,len(array[i])):
9         point = rs.CreatePoint(array[i][j].X, array[i][j].Y, brightness[b]*500)
10        points.append(point)
11        b = b+1
12    line = rs.AddCurve(points)
13    lines.append(line)
```

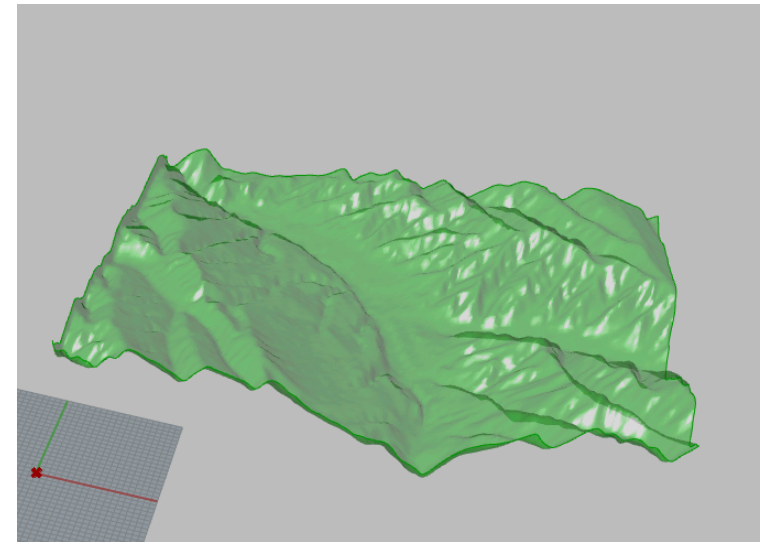
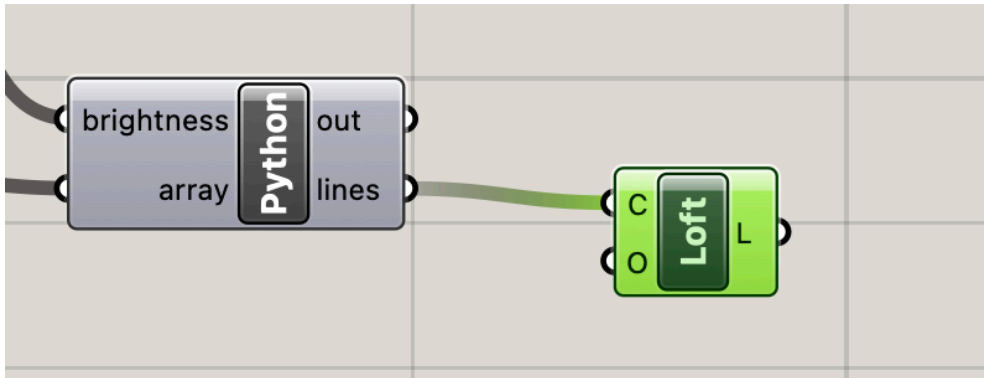




# Array of lines is final output



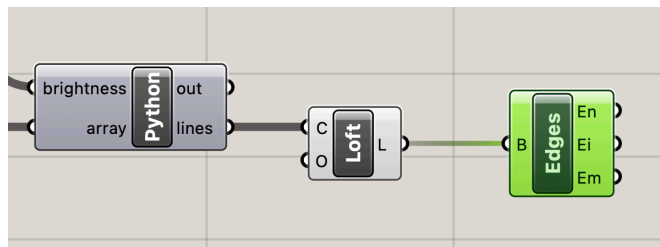
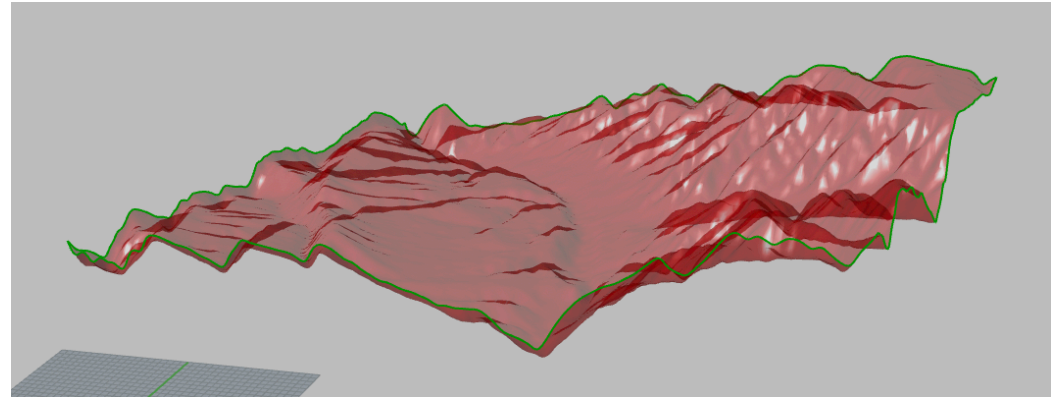
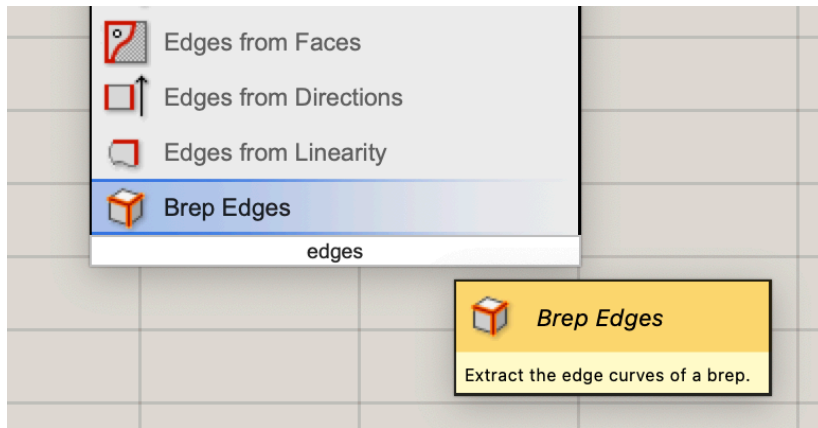
# Loft to create surface from lines



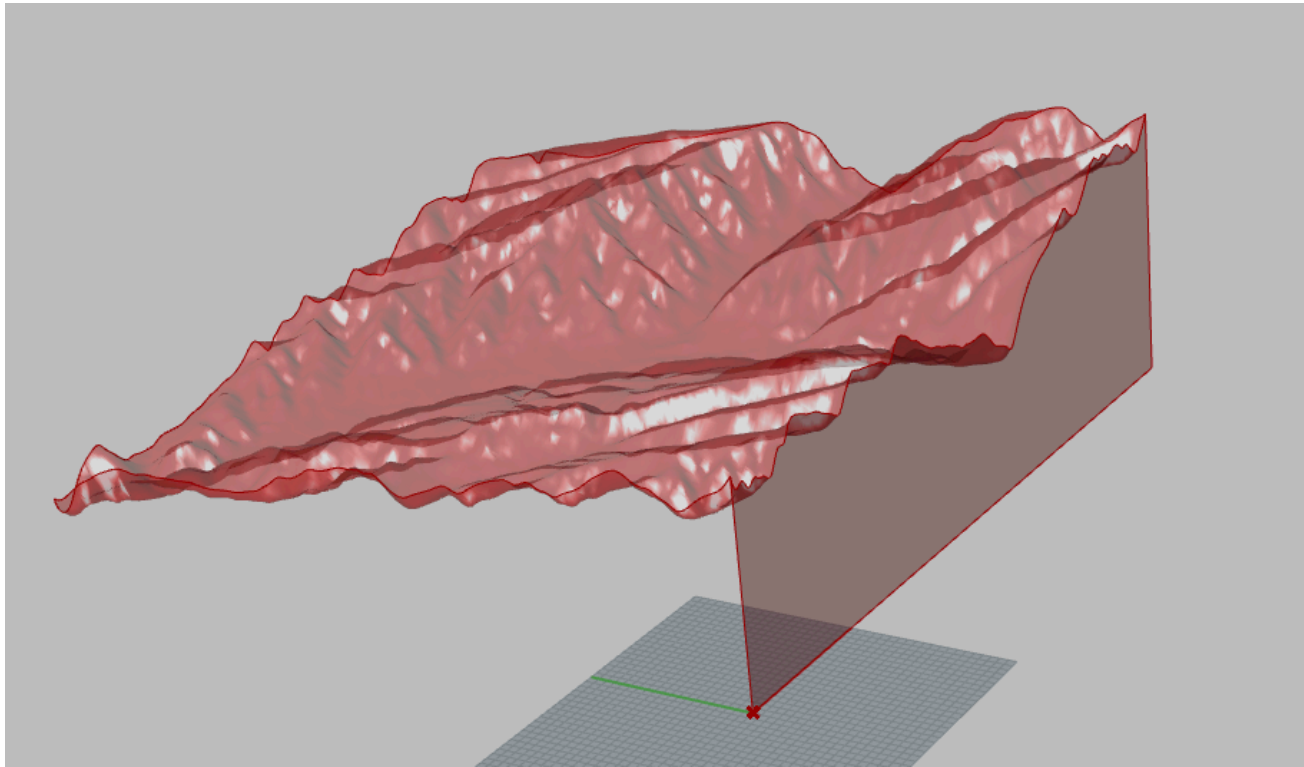
questions?

Creating a solid

# Get Edges of Surface



# Create A Surface Under Each Edge



# Get Edge End Points

```
1 import rhinoscriptsyntax as rs
2
3 surfaces = []
4 for i in range (len(edges)):
5     lines=[]
6     point0 = rs.CurveEndPoint(edges[i])
7     point1 = rs.CurveStartPoint(edges[i])
```

<https://developer.rhino3d.com/api/RhinoScriptSyntax/#curve-CurveEndPoint>

## CurveEndPoint

```
CurveEndPoint(curve_id, segment_index=-1)
```

Returns the end point of a curve object

**Parameters:**

`curve_id` (guid): identifier of the curve object

`segment_index` (number, optional): the curve segment index if `curve\_id` identifies a polycurve

**Returns:**

`point`: The 3d endpoint of the curve if successful.

`None`: on error

## CurveStartPoint

```
CurveStartPoint(curve_id, segment_index=-1, point=None)
```

Returns the start point of a curve object

**Parameters:**

`curve_id` (guid): identifier of the curve object

`segment_index` (number, optional): the curve segment index if `curve\_id` identifies a polycurve

`point` (point, optional): new start point

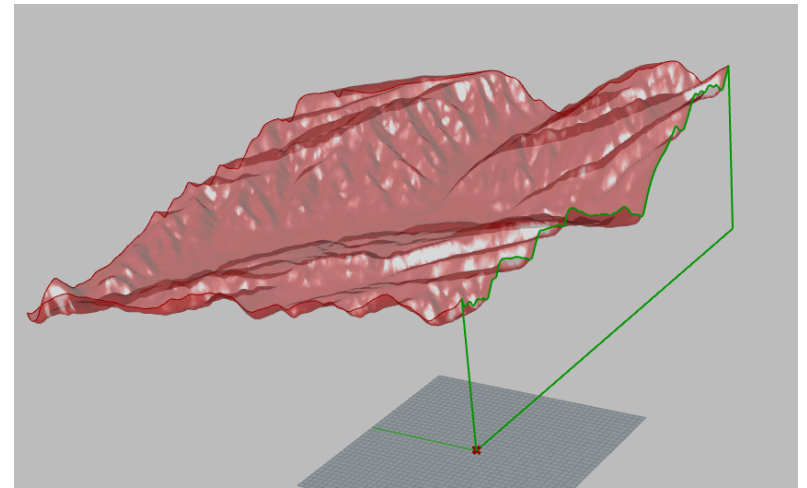
**Returns:**

`point`: The 3D starting point of the curve if successful.



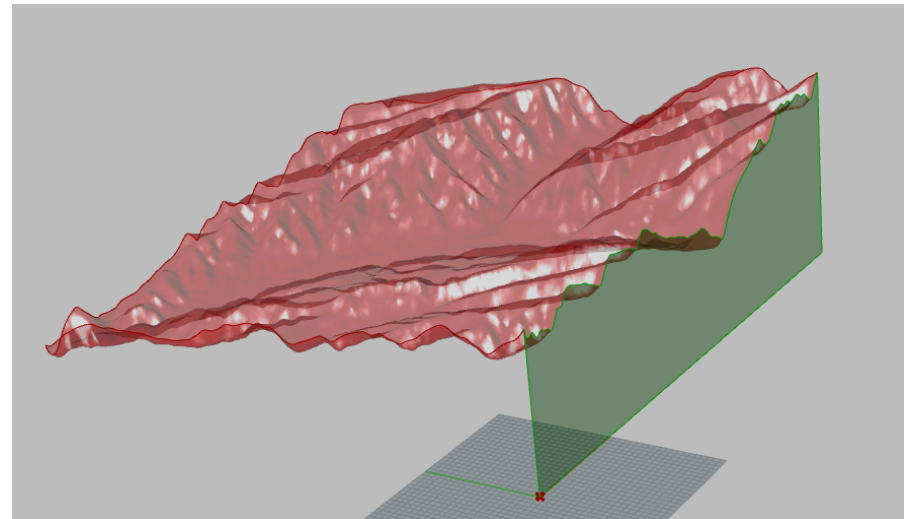
# Create Edges for Your Surfaces

```
1 import rhinoscriptsyntax as rs
2
3 surfaces = []
4 for i in range (len(edges)):
5     lines=[]
6     point0 = rs.CurveEndPoint(edges[i])
7     point1 = rs.CurveStartPoint(edges[i])
8     point2 = rs.CreatePoint(point0.X, point0.Y,0)
9     point3 = rs.CreatePoint(point1.X, point1.Y,0)
10
11     lines.append(edges[i])
12     lines.append(rs.AddLine(point0,point2))
13     lines.append(rs.AddLine(point2,point3))
14     lines.append(rs.AddLine(point3,point1))
15
```

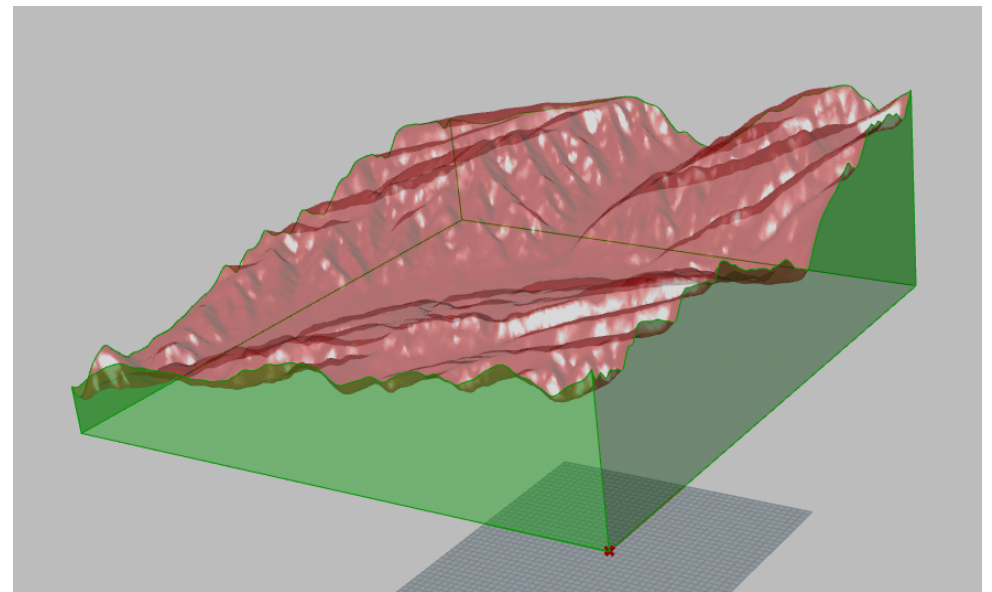
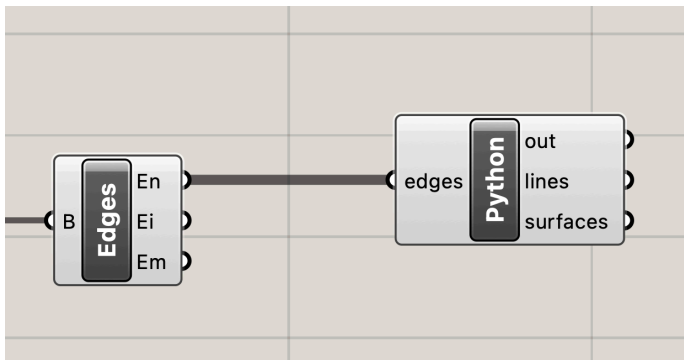


# Create Surfaces from Edge Curves

```
1 import rhinoscriptsyntax as rs
2
3 surfaces = []
4 for i in range (len(edges)):
5     lines=[]
6     point0 = rs.CurveEndPoint(edges[i])
7     point1 = rs.CurveStartPoint(edges[i])
8     point2 = rs.CreatePoint(point0.X, point0.Y,0)
9     point3 = rs.CreatePoint(point1.X, point1.Y,0)
10
11     lines.append(edges[i])
12     lines.append(rs.AddLine(point0,point2))
13     lines.append(rs.AddLine(point2,point3))
14     lines.append(rs.AddLine(point3,point1))
15
16     surface = rs.AddEdgeSrf(lines)
17     surfaces.append(surface)
```

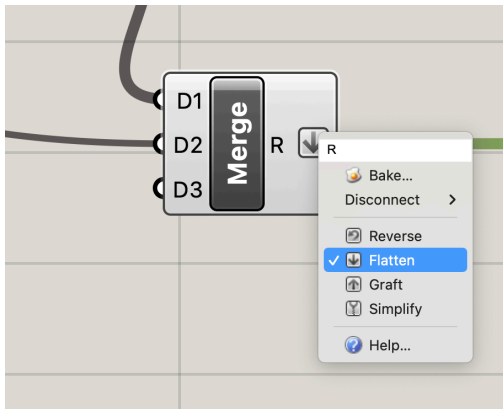


# Create Surfaces from Edge Curves

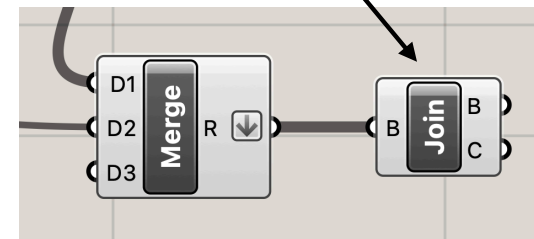
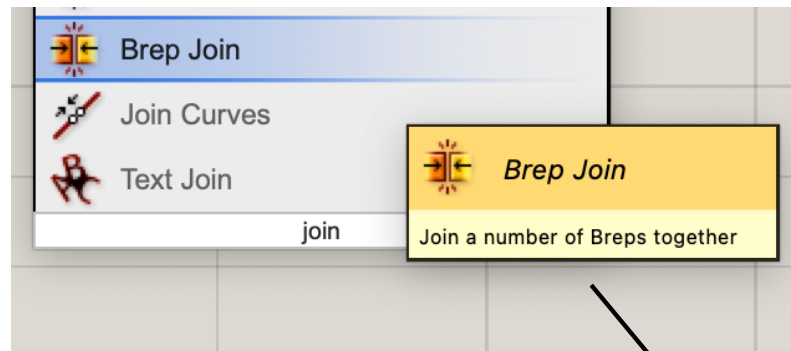


questions?

# Join Top and Side Surfaces

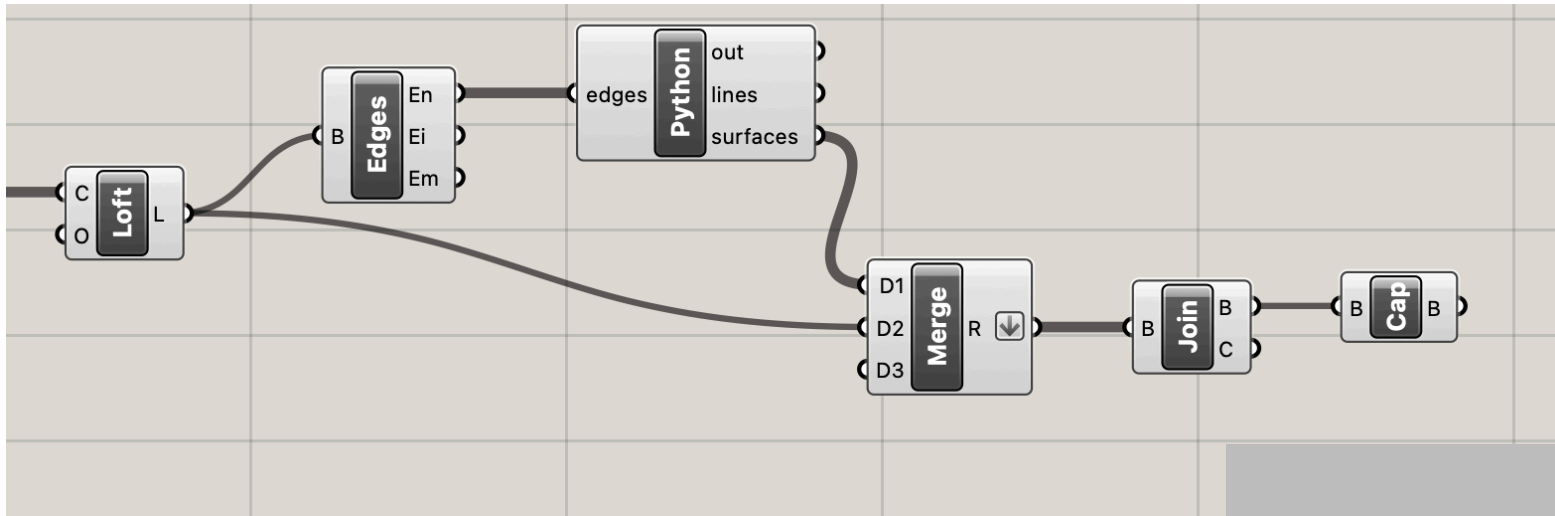


Use Merge to  
Create a list of all surfaces  
Flatten list to create single list

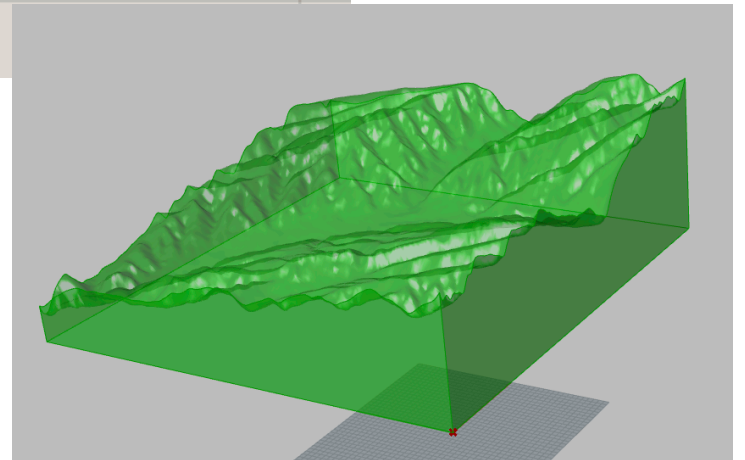


Join all surfaces using Brep Jpoin

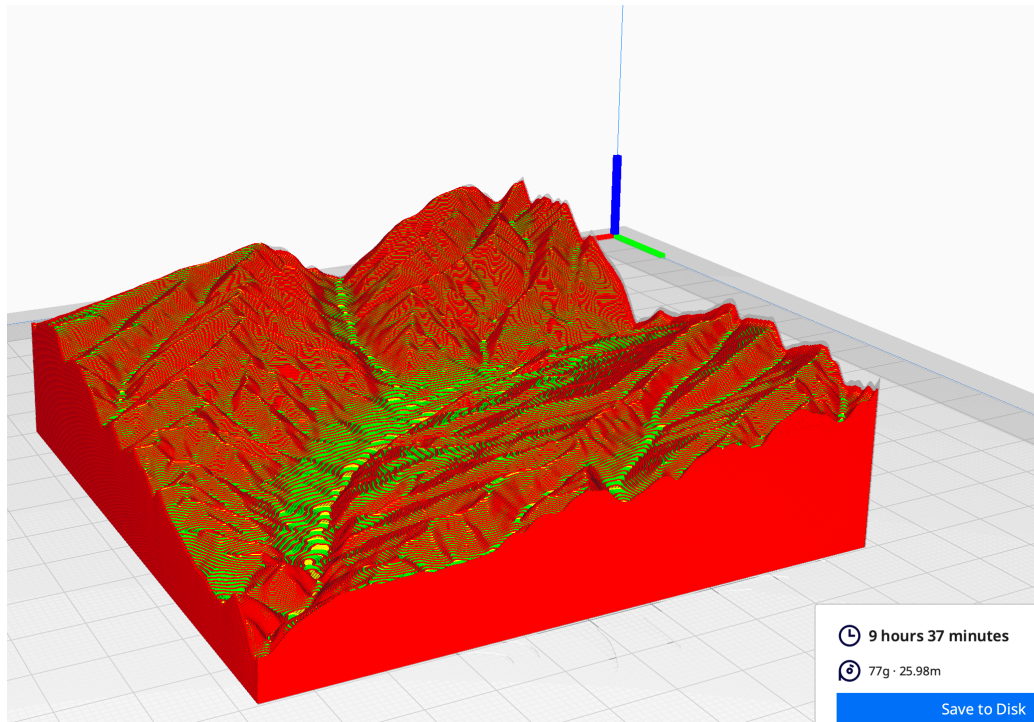
# Use Cap to Create a Bottom



Result is solid



# Bake, Slice, and Print





questions?

# Working with Data, Stepping Back

# Finding Data

- Data based design: data first then design
- Data can be hard to find.
  - Easier to find data:
    - Non-human data. ie: weather, geography, animal tracking, etc.
    - Population level data. ie: economic data, urban data, etc.
    - Anonymized and summary data.
  - Particularly hard to find data:
    - Personal health data.
    - The more sensitive the data, the harder to find.
- Use scholarly and governmental sources. (<https://scholar.google.com/>)  
Google probably won't serve you well.

# Interpreting & Cleaning Data

- Data is in weird maybe proprietary formats, often hard to make sense of, incomplete, poorly structured and inconsistent
- Understanding data can significant take time and effort
- Cleaning data is an important part of the process

STATION	DATE	REPORT_TYPE	SOURCE	AWND	BackupDirection	BackupDistance	BackupDista	BackupElements	BackupEleva	BackupEleva	BackupEquip	BackupLatitu	BackupLongi	BackupNam< CDS	CLDD
72365023050	2023-01-01T00:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T01:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T02:00:00	FM-12		4				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T02:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T03:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T04:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T05:00:00	FM-12		4				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T05:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T06:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T07:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T08:00:00	FM-12		4				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T08:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T09:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T10:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T11:00:00	FM-12		4				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T11:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T12:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T13:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T14:00:00	FM-12		4				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T14:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T15:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T16:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T17:00:00	FM-12		4				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T17:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T18:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T19:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T20:00:00	FM-12		4				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T20:43:00	FM-16		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T20:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T21:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T22:50:00	FM-16		6				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	
72365023050	2023-01-01T22:52:00	FM-15		7				PRECIP, TEMP, SNOWFALL	5324		SRG, MMTS,	35.057778	-106.61667	ALBUQUERQUE INTL AP COOP	

DATE	Temperature	Humidity
2023-01-01T00:52:00	40	77
2023-01-01T01:52:00	40	77
2023-01-01T02:00:00	40	77
2023-01-01T02:52:00	38	79
2023-01-01T03:52:00	40	77
2023-01-01T04:52:00	41	73
2023-01-01T05:00:00	41	73
2023-01-01T05:52:00	40	79
2023-01-01T06:52:00	39	82
2023-01-01T07:52:00	36	89
2023-01-01T08:00:00	36	89
2023-01-01T08:52:00	40	79
2023-01-01T09:52:00	41	79
2023-01-01T10:52:00	43	74
2023-01-01T11:00:00	43	74
2023-01-01T11:52:00	44	71
2023-01-01T12:52:00	46	66
2023-01-01T13:52:00	48	61
2023-01-01T14:00:00	48	61
2023-01-01T14:52:00	50	57

# Interpreting & Cleaning Data



City of Albuquerque  
Department of Technology and Innovation  
Data Management

## Data.cabq.gov Core Metadata Requirements

### Contact Information

Name	Mark Leech
Department/Division	Albuquerque Police Department
Phone	505-768-3731
Email	<a href="mailto:mleech@cabq.gov">mleech@cabq.gov</a>

### What Does this Dataset Describe?

Dataset Title	APD_Incidents
Short Description	Location of calls for service by APD
<b>Full Non-Technical Description</b>	
This dataset contains the block location, case number description and date of calls for service received by APD that have been entered into the Computer Aided Dispatch (CAD) system and subsequently closed. No personally identifiable information (PII) is released. This dataset is a rolling 180 days of incidents.	

```
{
  "attributes": {
    "OBJECTID": 61868319,
    "BlockAddress": "WYOMING BL NE / MARQUETTE AV NE",
    "IncidentType": "SHOTS FIRED",
    "ReportDateTime": "2023-04-30 00:32:01"
  },
  "geometry": {
    "x": -11861135.0196,
    "y": 4175256.863499999
  }
},
{
  "attributes": {
    "OBJECTID": 61868320,
    "BlockAddress": "CENTRAL AV NE / CHAMA ST NE",
    "IncidentType": "THEFT/FRAUD/EMBE",
    "ReportDateTime": "2023-04-30 00:36:35"
  },
  "geometry": {
    "x": -11862913.8325,
    "y": 4174167.8104000017
  }
},
{
  "attributes": {
    "OBJECTID": 61868321,
    "BlockAddress": "200 BLOCK EUBANK BL SE",
    "IncidentType": "DISTURBANCE",
    "ReportDateTime": "2023-04-30 00:00:20"
  },
  "geometry": {
    "x": -11859142.418,
    "y": 4173502.2797999978
  }
},
}
```



questions?



# Thank you!

CS 491 and 591

Professor: Leah Buechley

[https://handandmachine.cs.unm.edu/classes/Computational\\_Fabrication\\_Spring2021/](https://handandmachine.cs.unm.edu/classes/Computational_Fabrication_Spring2021/)