

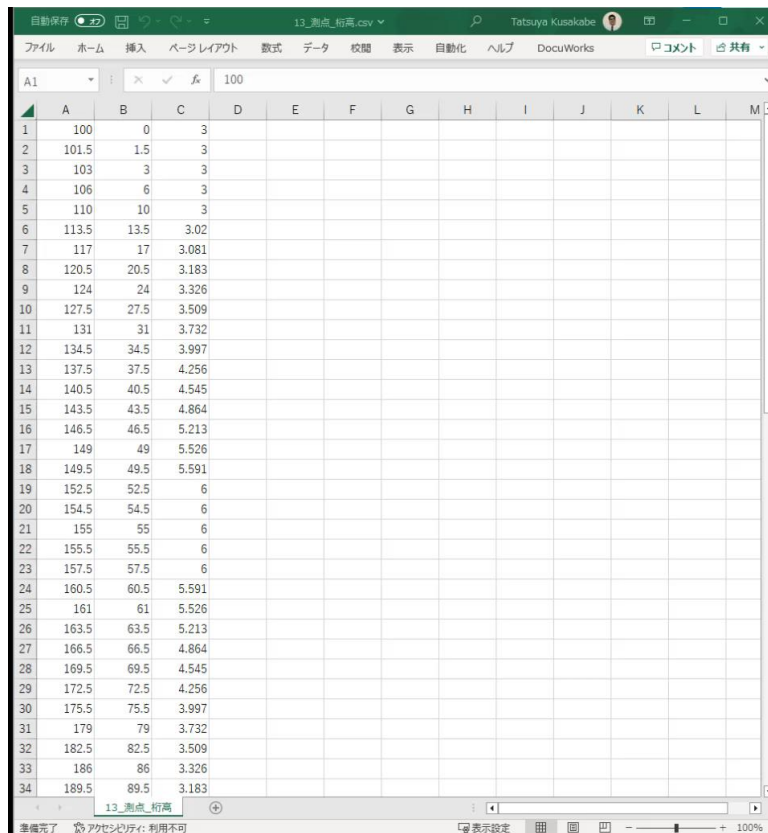


Dynamo サンプル解説

橋梁モデルの自動作成

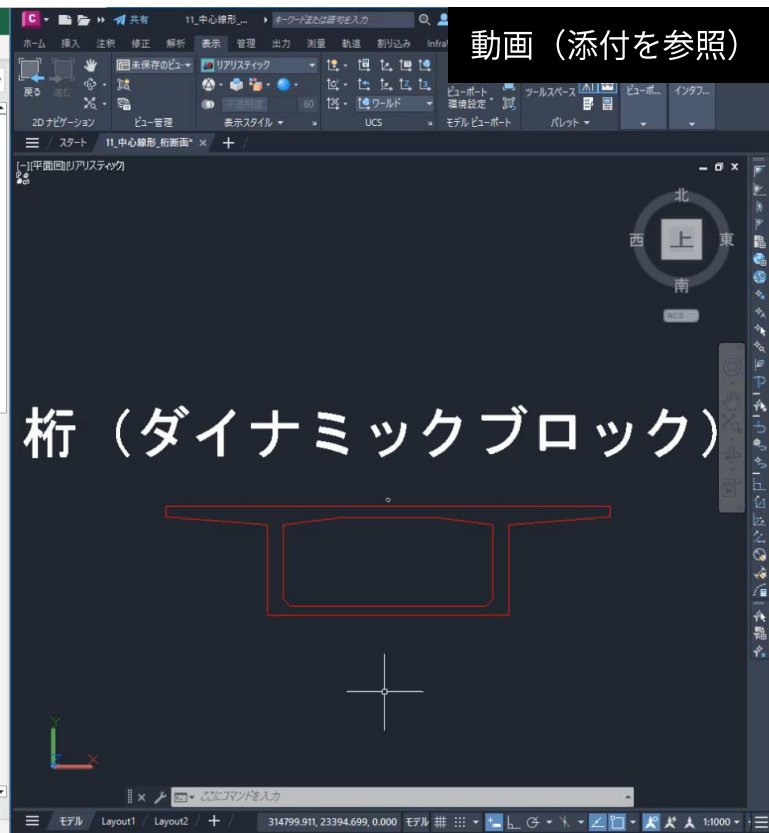
Dynamo サンプル

測点と桁高の csv と、桁断面のブロックから、橋梁モデルを自動作成



The image shows a spreadsheet application with a CSV file named '13_測点_桁高.csv'. The data is organized into columns A through M and rows 1 through 34. The first three columns (A, B, C) contain numerical data representing bridge measurements. The remaining columns (D through M) are empty.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	100	0	3										
2	101.5	1.5	3										
3	103	3	3										
4	106	6	3										
5	110	10	3										
6	113.5	13.5	3.02										
7	117	17	3.081										
8	120.5	20.5	3.183										
9	124	24	3.326										
10	127.5	27.5	3.509										
11	131	31	3.732										
12	134.5	34.5	3.997										
13	137.5	37.5	4.256										
14	140.5	40.5	4.545										
15	143.5	43.5	4.864										
16	146.5	46.5	5.213										
17	149	49	5.526										
18	149.5	49.5	5.591										
19	152.5	52.5	6										
20	154.5	54.5	6										
21	155	55	6										
22	155.5	55.5	6										
23	157.5	57.5	6										
24	160.5	60.5	5.591										
25	161	61	5.526										
26	163.5	63.5	5.213										
27	166.5	66.5	4.864										
28	169.5	69.5	4.545										
29	172.5	72.5	4.256										
30	175.5	75.5	3.997										
31	179	79	3.732										
32	182.5	82.5	3.509										
33	186	86	3.326										
34	189.5	89.5	3.183										



先ほどの Dynamo サンプルの中身を、細かく解説

- 含まれる内容

- csv との連携
- Civil 3D との連携
 - 平面線形、縦断線形、ブロック
- リストの編集
- 図形（ジオメトリ）の作成
- 座標系の作成・編集
- パッケージの活用
- Python スクリプトの活用

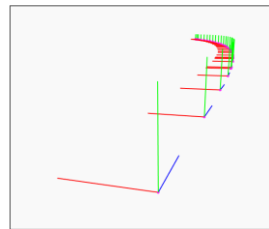
- データセット・デモ動画

- <https://autode.sk/3NerorC>

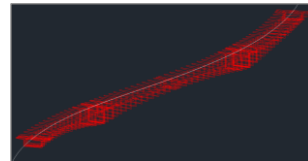


- データセットは、下記に分かれています

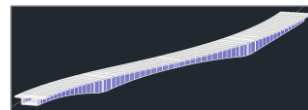
- 練習 1) Civil 3D の縦断線形に沿って、座標系を作成
 - 解説 1) Civil 3D 平面線形 & 縦断線形 - 取得
 - 解説 2) Civil 3D 平面線形 & 縦断線形 - 計算
 - 解説 3) Excel / csv との入出力、リストの編集
 - 解説 4) 座標系の編集



-
- 練習 2) Civil 3D の縦断線形に沿って、ブロックを配置
 - 解説 5) AutoCAD ブロック定義
 - 解説 6) AutoCAD ブロック参照



-
- 練習 3) 配置したブロックからソリッドを作成
 - 解説 7) Civil3DToolkit パッケージの活用：
ブロック参照を Dynamo ポリラインに変換
 - 解説 8) Python スクリプトの活用：
Dynamo ポリラインをソリッドに変換



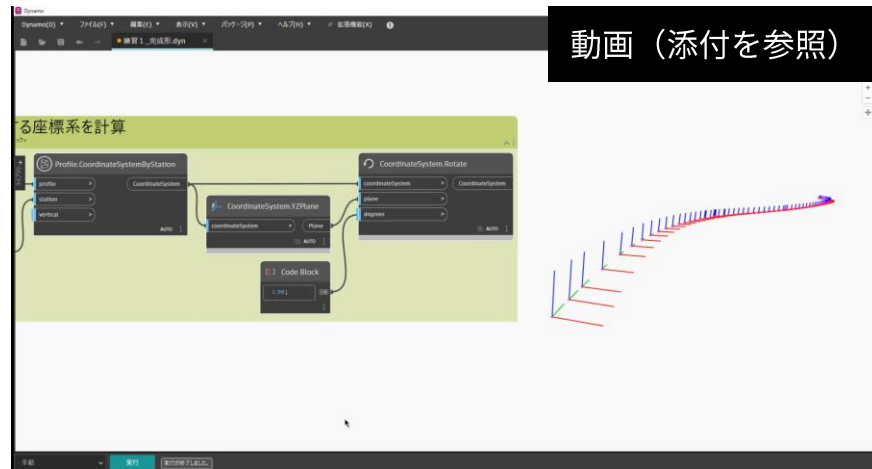
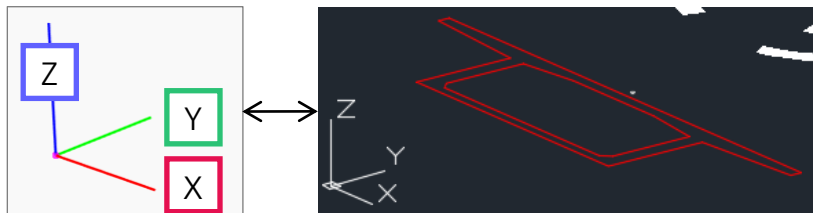
練習 1) Civil 3D の縦断線形に沿って、座標系を作成

完成形を先にお見せします

- 手順
 - Civil 3D から、縦断線形を取得
 - csv から、ブロックを配置する測点を取得

	測点		桁高
	A	B	C
1	100	0	3
2	101.5	1.5	3
3	103	3	3

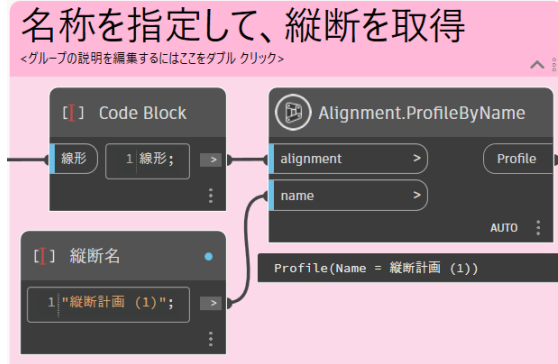
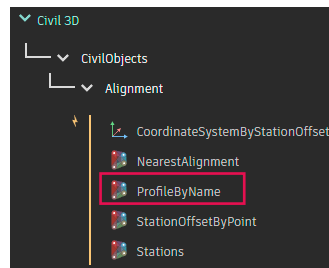
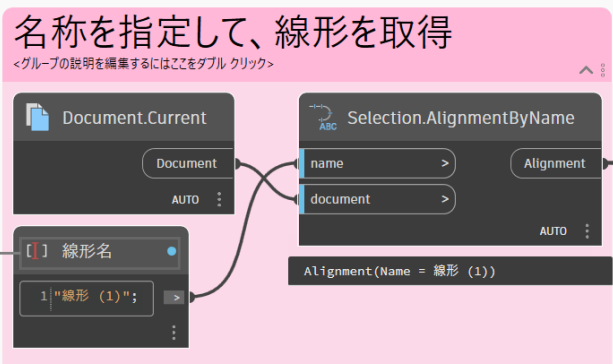
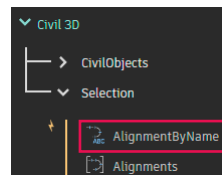
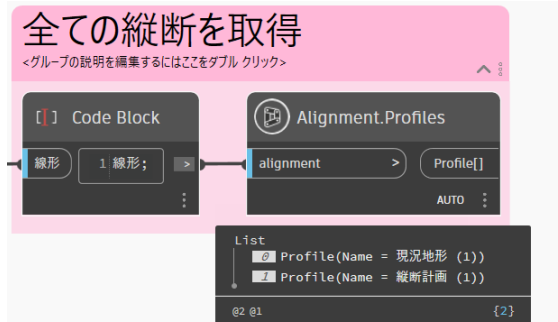
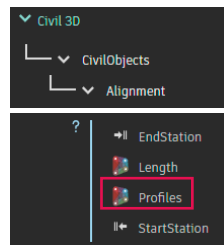
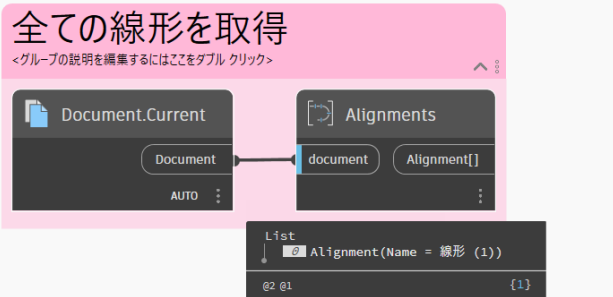
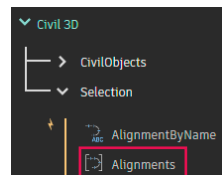
- 縦断線形と測点に合わせ、座標系を配置
- 座標系を回転
 - 座標系の XY 平面 = ブロックの XY 平面



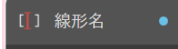
動画（添付を参照）

解説 1) Civil 3D 平面線形 & 縦断線形 - 取得

平面線形 や 縦断線形 を取得するには...

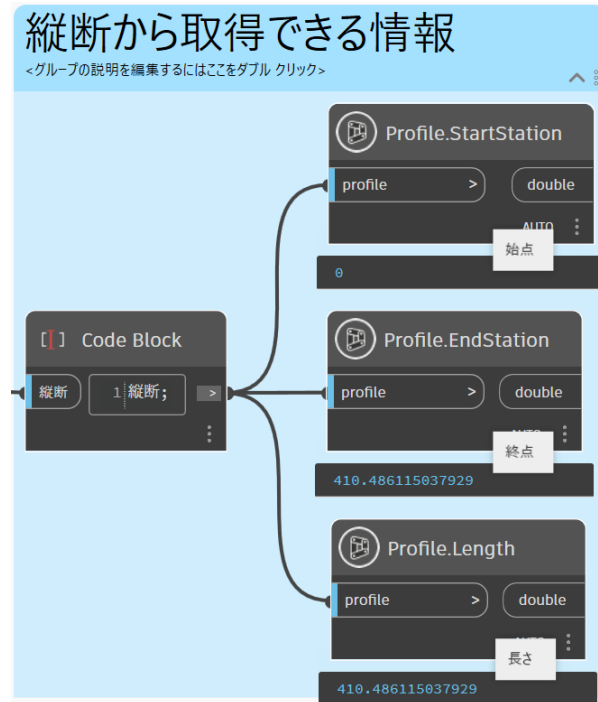
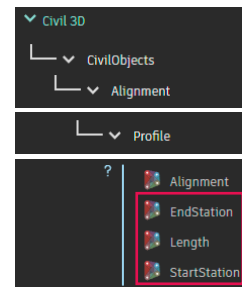
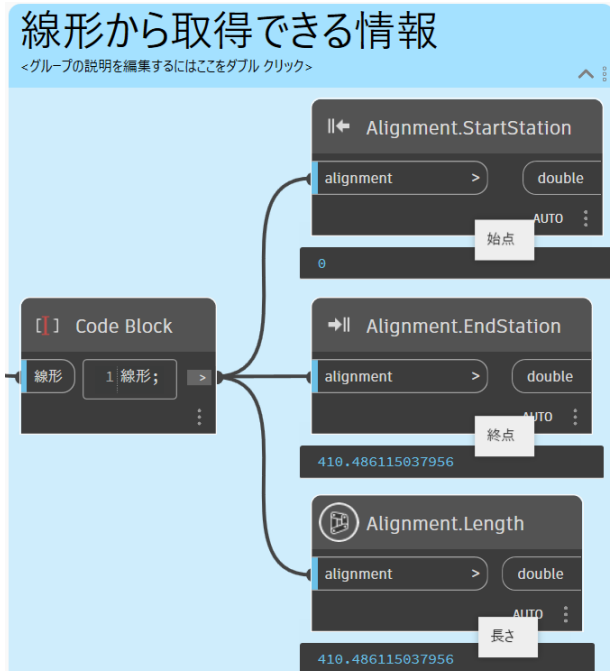
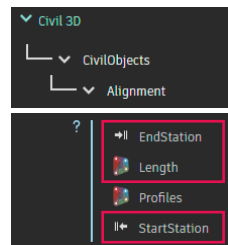


Code Block から名称を変更
→ ● マークがつく



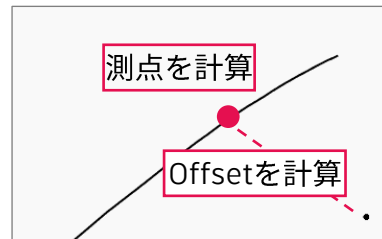
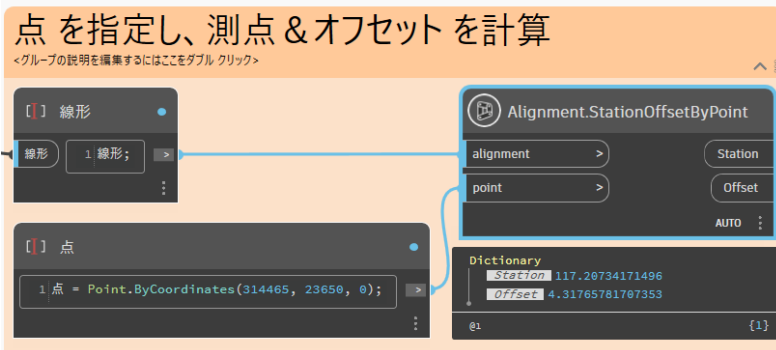
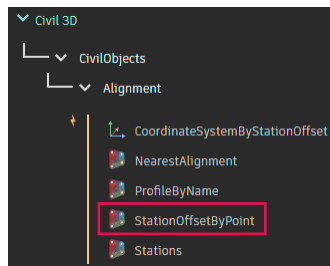
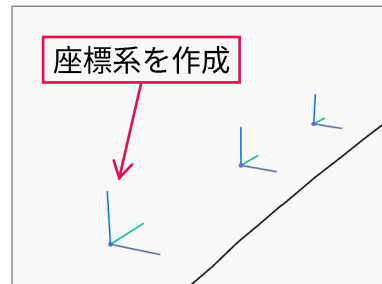
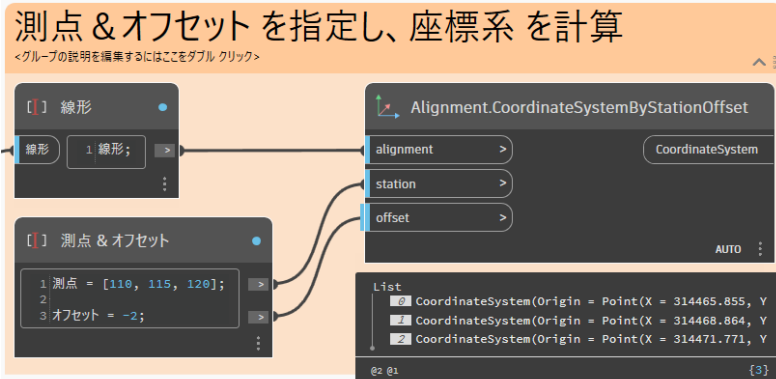
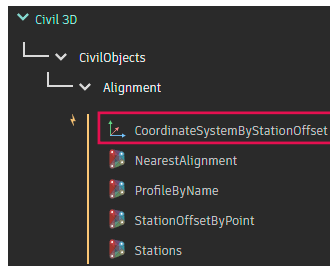
解説 1) Civil 3D 平面線形 & 縦断線形 - 取得

平面線形 や 縦断線形 から、そのまま抜き出せる情報は...



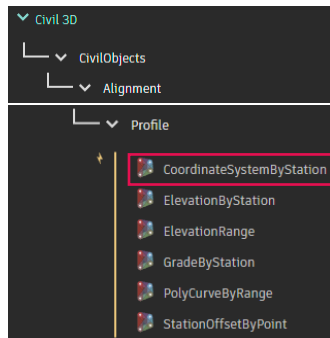
解説 2) Civil 3D 平面線形 & 縦断線形 - 計算

平面線形 から、計算して求められる情報は...



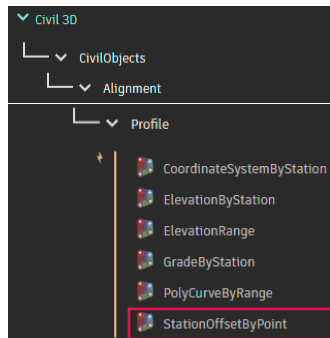
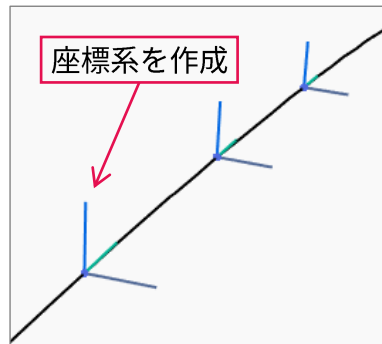
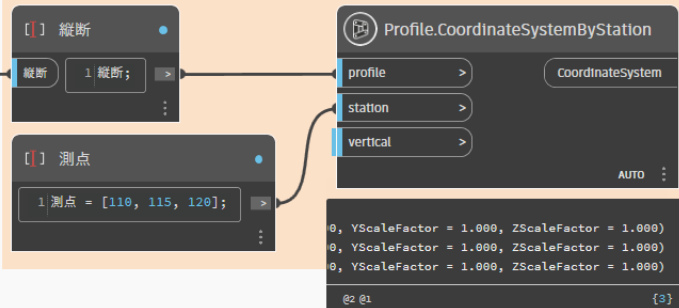
解説 2) Civil 3D 平面線形 & 縦断線形 - 計算

縦断線形 から、計算して求められる情報は...



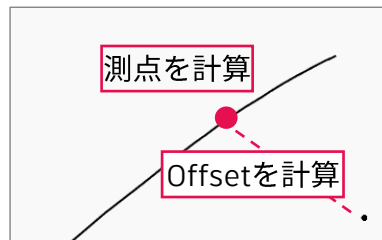
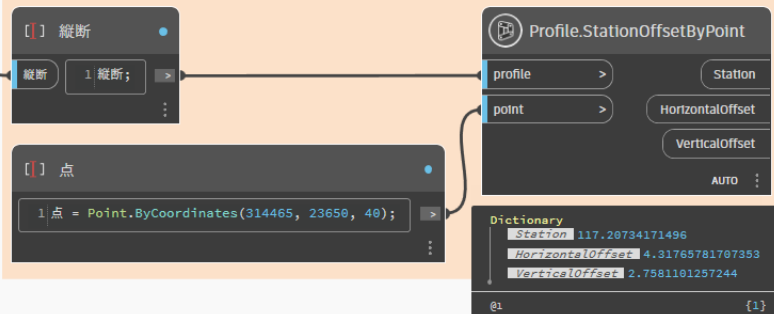
測点を指定し、座標系を計算

<グループの説明を編集するにはここをダブルクリック>



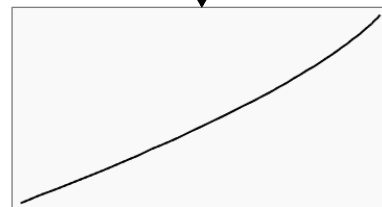
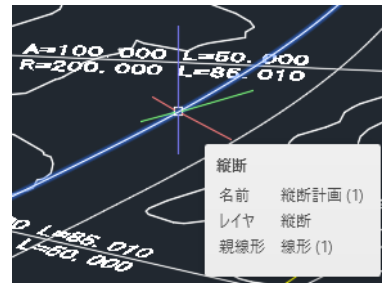
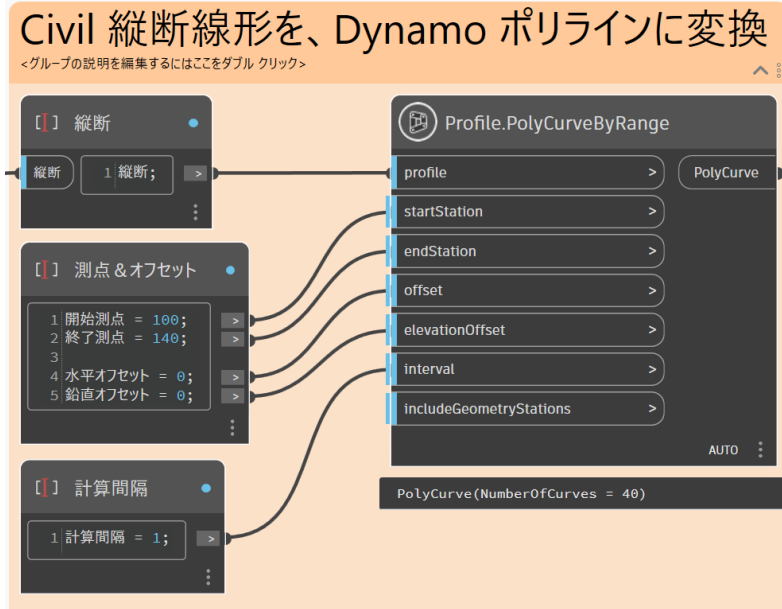
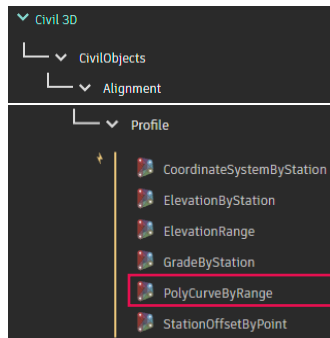
点を指定し、測点 & オフセットを計算

<グループの説明を編集するにはここをダブルクリック>



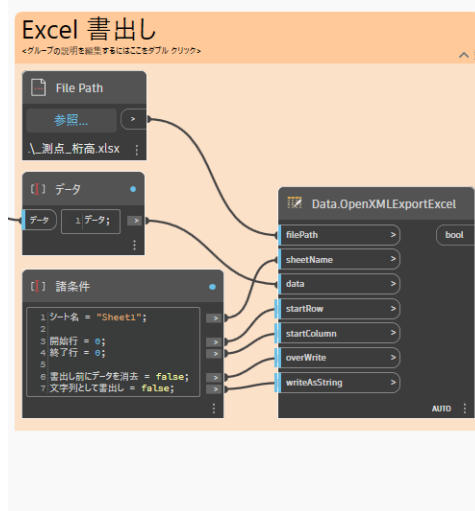
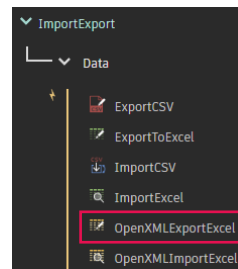
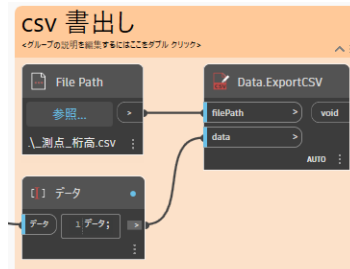
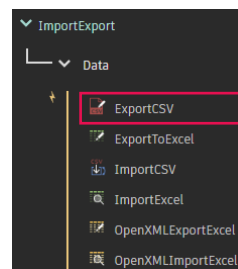
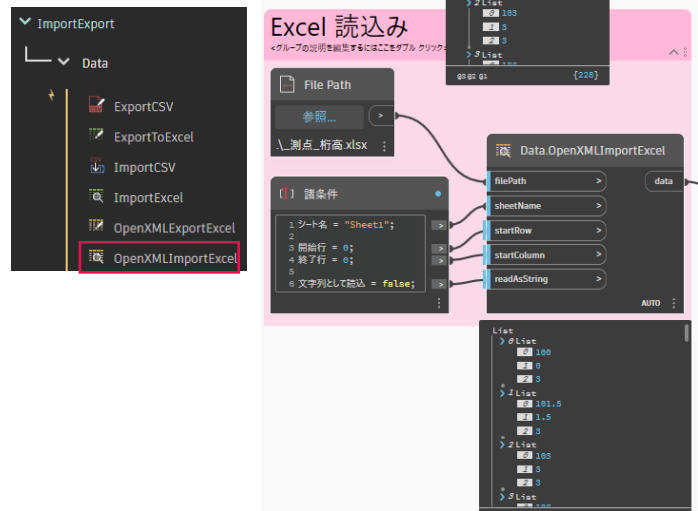
解説 2) Civil 3D 平面線形 & 縦断線形 - 計算

縦断線形 は、Dynamo ジオメトリに変換できる



解説3) Excel / csv との連携、リストの編集

Excel や csv から、データを入出力するには... (復習も兼ねて)

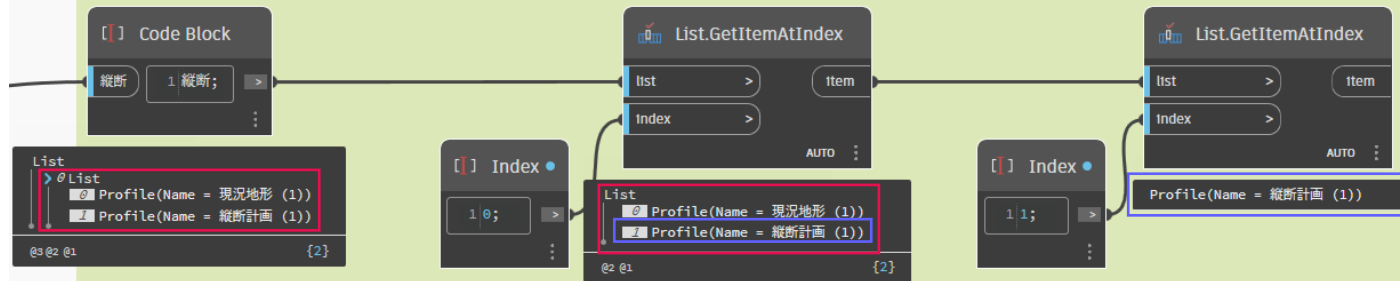


解説 3) Excel / csv との連携、リストの編集

リストから、特定の要素を抽出するには... (Code Block が便利！)

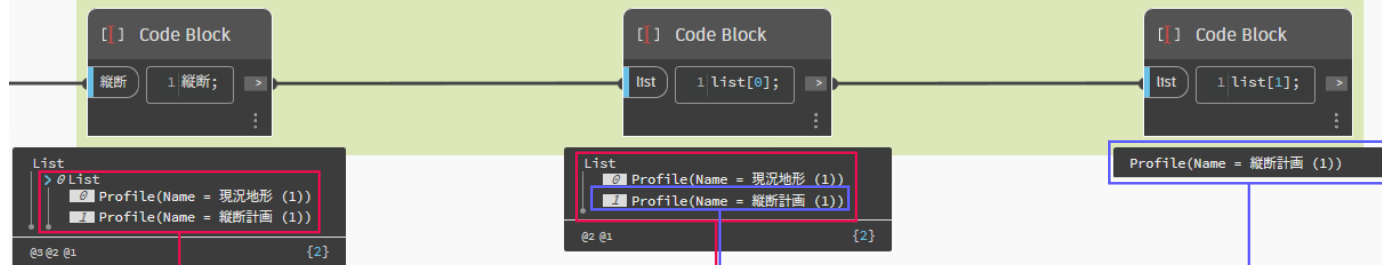
リストの中から縦断計画を取得 (List.GetItemAtIndex 関数で)

<グループの説明を編集するにはここをダブル クリック>



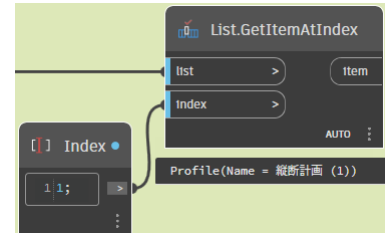
リストの中から縦断計画を取得 (Code Block で)

<グループの説明を編集するにはここをダブル クリック>



0 番目の要素を取得

1 番目の要素を取得



解説 3) Excel / csv との連携、リストの編集

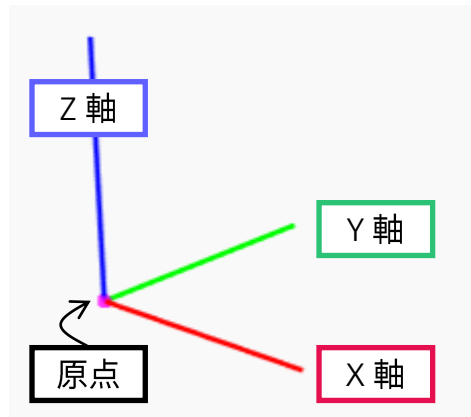
リストから、特定の要素を抽出するには... (Code Block が便利！)



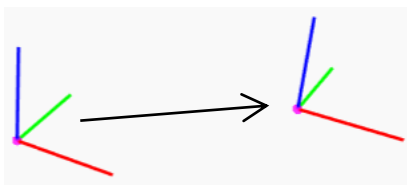
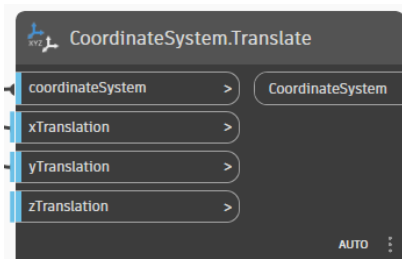
解説 4) 座標系の編集

“座標系” とは？（復習も兼ねて）

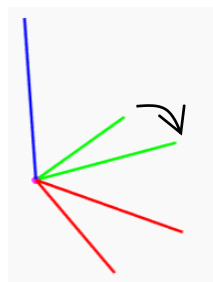
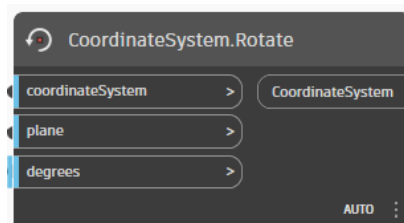
座標系：位置（原点）と
方向（X, Y, Z 軸）の
組合せ



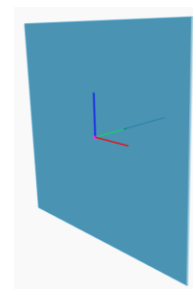
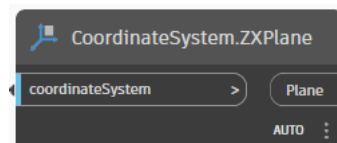
移動したり



回転したり

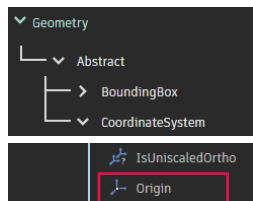


断面を求めたり



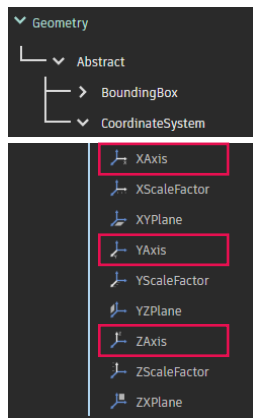
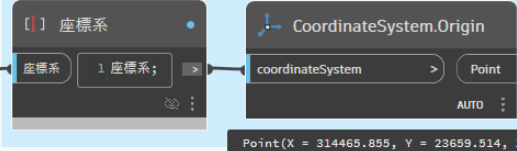
解説 4) 座標系の編集

座標系 から、そのまま抜き出せる情報は...



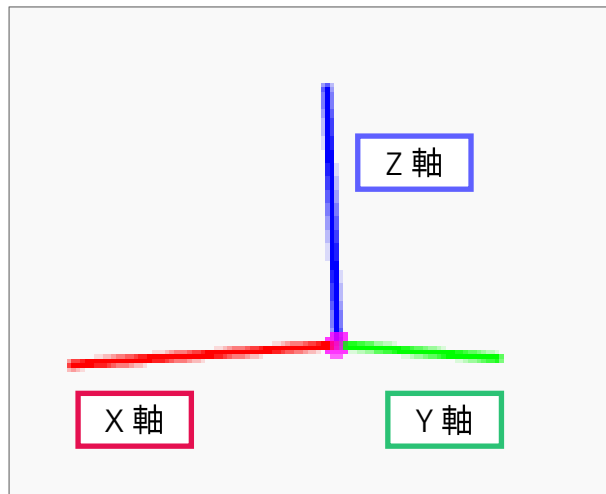
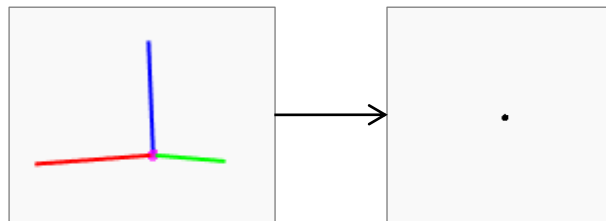
座標系の原点を取得

<グループの説明を編集するにはここをダブルクリック>



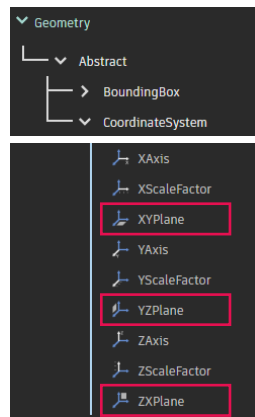
座標系の軸ベクトルを取得

<グループの説明を編集するにはここをダブルクリック>



解説 4) 座標系の編集

座標系 から、そのまま抜き出せる情報は...

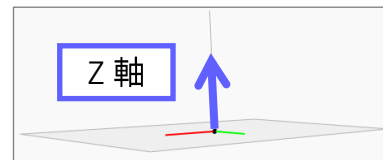


座標系の平面を取得

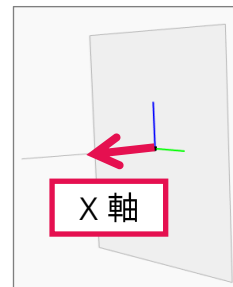
<グループの説明を編集するにはここをダブルクリック>



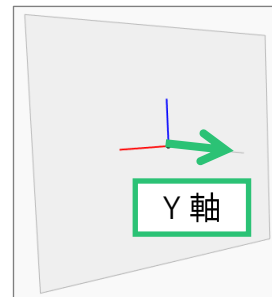
XYPlane :
Z 軸に直交
する平面



YZPlane :
X 軸に直交
する平面

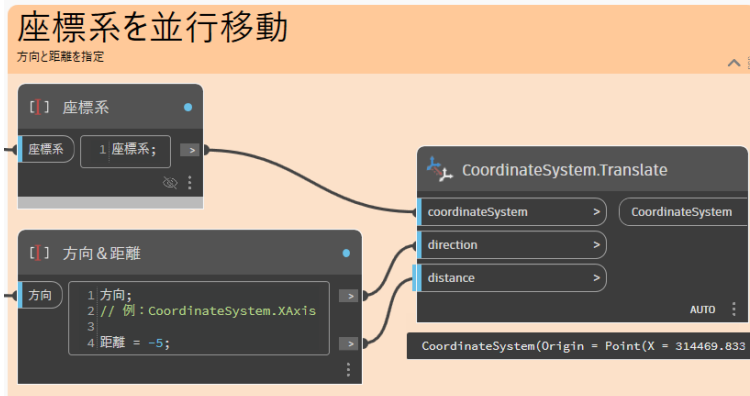
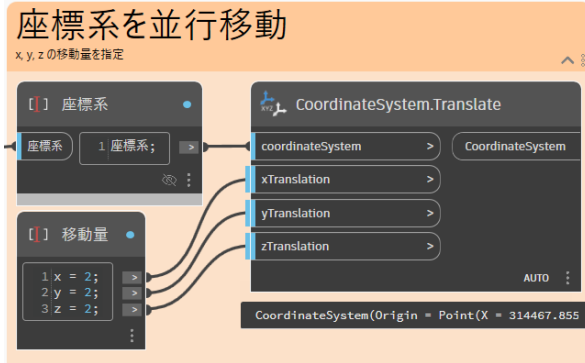
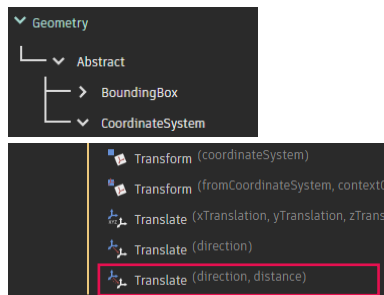
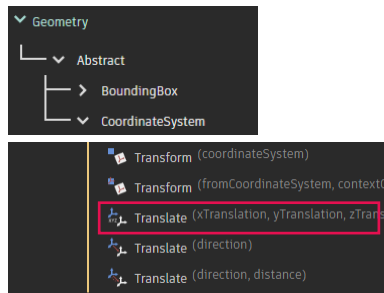


ZXPlane :
Y 軸に直交
する平面



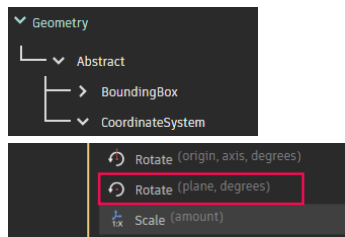
解説 4) 座標系の編集

座標系 は、動かすことができる



解説 4) 座標系の編集

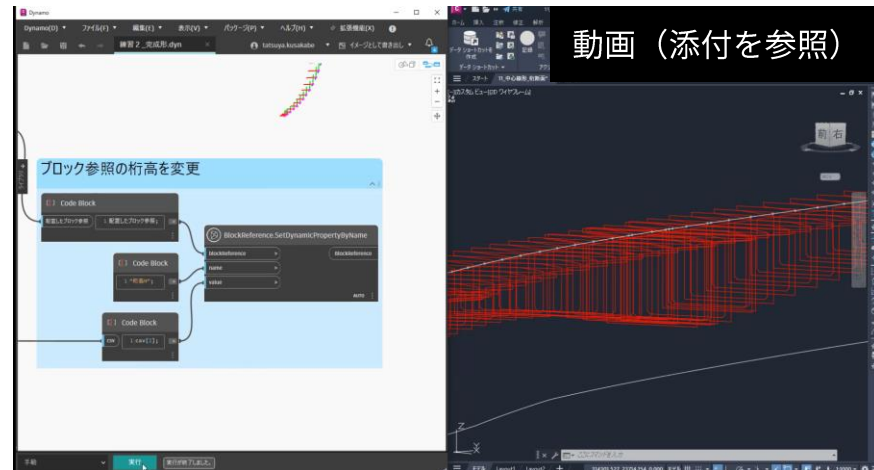
座標系 は、動かすことができる



練習 2) Civil 3D の縦断線形に沿って、ブロックを配置

完成形を先にお見せします

- 手順
 - Civil 3D から、ブロック定義を取得
 - Civil 3D に、ブロック参照を作成
 - ブロック参照のパラメータを変更
 - ・ ダイナミックブロックの活用



解説 5) AutoCAD ブロック定義

二つの“ブロック” ... “ブロック定義” と “ブロック参照”

ブロック定義

図形を組み合わせ、一まとめに登録

例：BLOCKSPALETTE の一覧



桁断面

路線記号

人孔記号

断面記号

...

要素

始点 X

始点 Y

終点 X

終点 Y

...

線分 1

xx

yy

xx

yy

...

...

ブロック参照

ブロック定義を参照し、CAD 上に配置

例：ブロック参照 オブジェクト



ブロック参照 1

ブロック参照 2

線分

...

ブロック定義

配置座標 X

配置座標 Y

...

桁断面

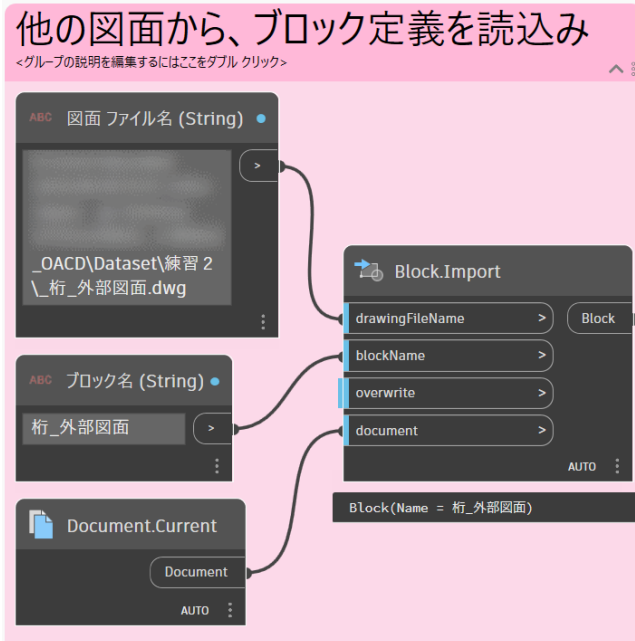
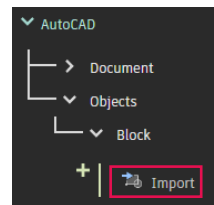
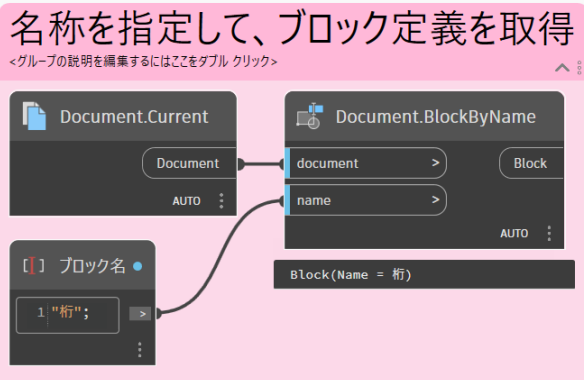
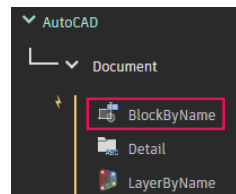
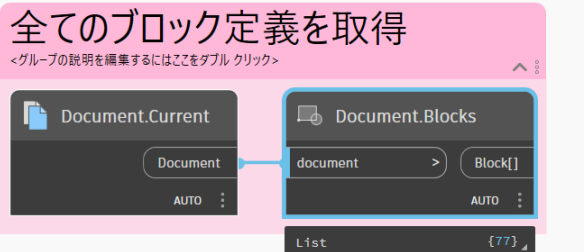
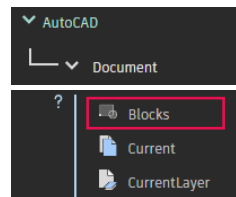
xx

yy

...

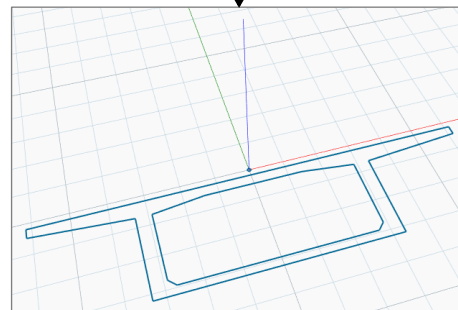
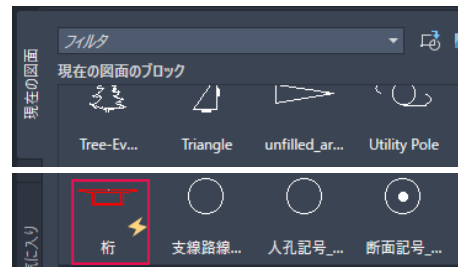
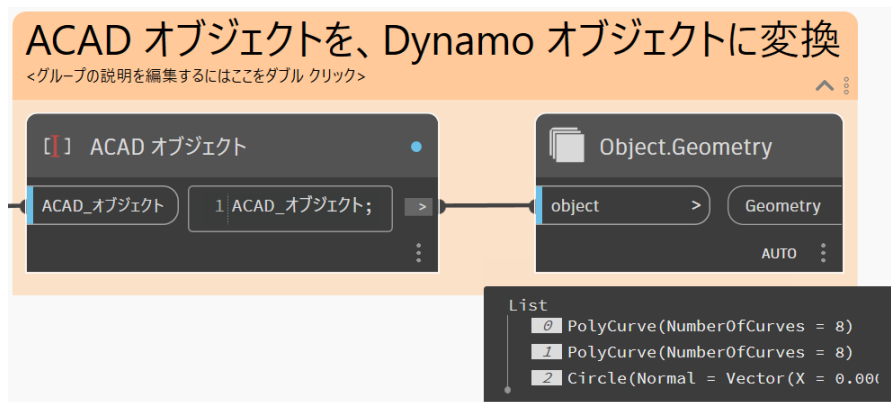
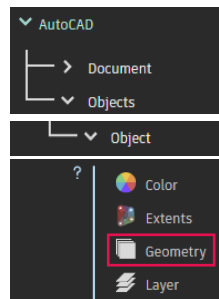
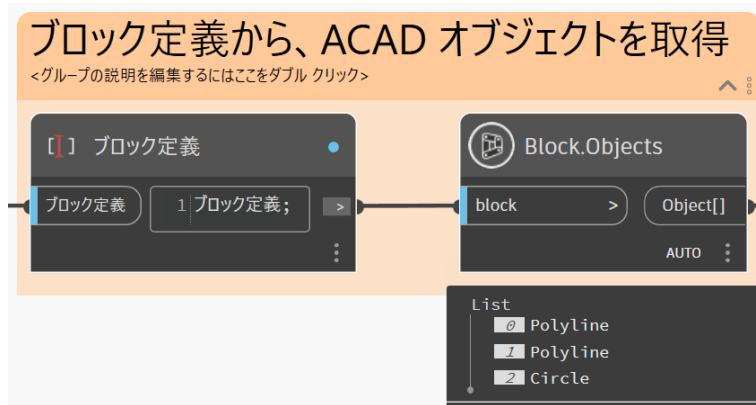
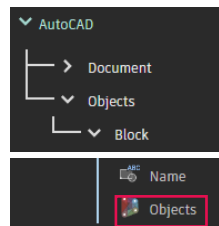
解説 5) AutoCAD ブロック定義

ブロック定義 を取得するには...



解説 5) AutoCAD ブロック定義

ブロック定義 は、Dynamo ジオメトリに変換できる



解説 6) AutoCAD ブロック参照

二つの“ブロック” ... “ブロック定義” と “ブロック参照”

ブロック定義

図形を組み合わせて、一まとめに登録

例：BLOCKSPALETTE の一覧



桁断面

路線記号

人孔記号

断面記号

...

要素

始点 X

始点 Y

終点 X

終点 Y

...

線分 1

xx

yy

xx

yy

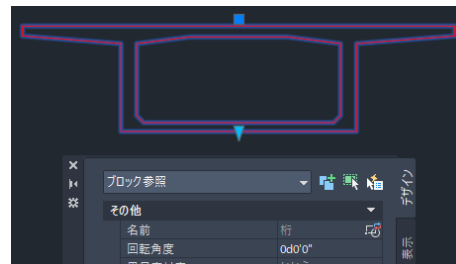
...

...

ブロック参照

ブロック定義を参照し、CAD 上に配置

例：ブロック参照 オブジェクト



ブロック参照 1

ブロック参照 2

線分

...

ブロック定義

配置座標 X

配置座標 Y

...

桁断面

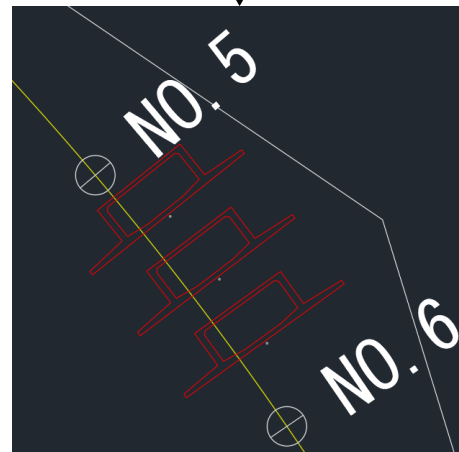
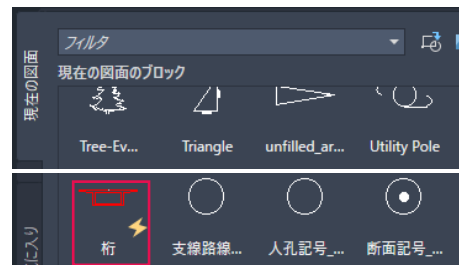
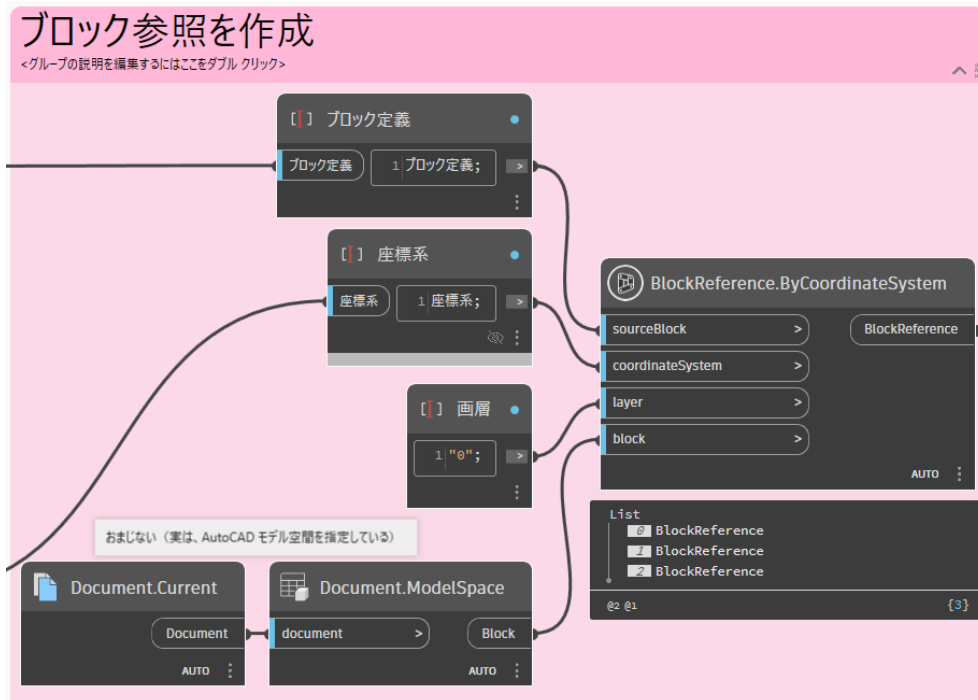
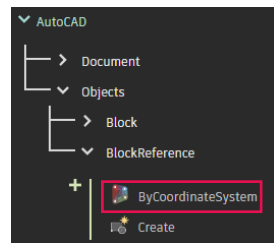
xx

yy

...

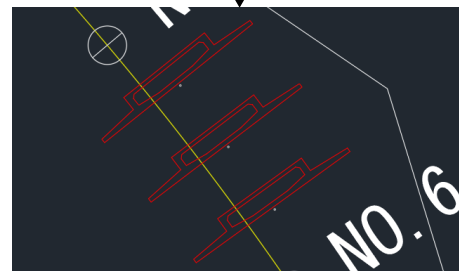
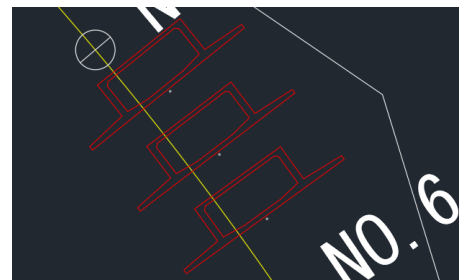
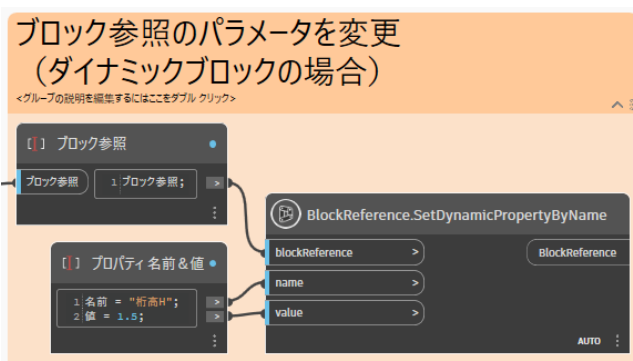
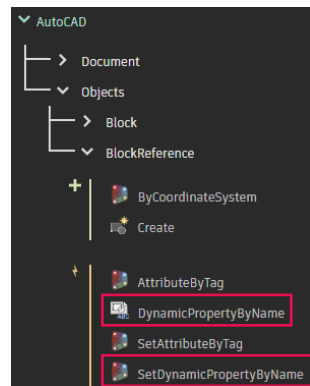
解説 6) AutoCAD ブロック参照

ブロック参照 を作成するには...



解説 6) AutoCAD ブロック参照

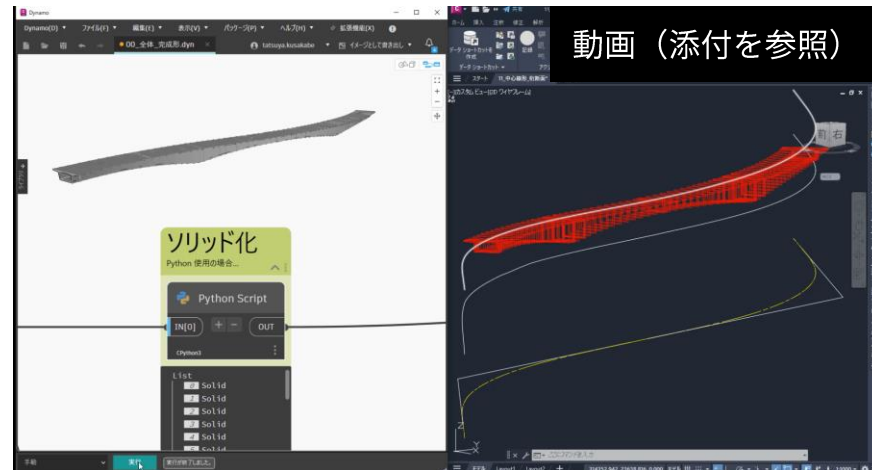
ブロック参照は、パラメトリックに形状を変えられる (ダイナミックブロックの場合)



練習 3) 配置したブロックから、ソリッドを作成

完成形を先にお見せします

- 手順
 - ブロック参照を Dynamo ポリラインに変換
 - ・ Civil3DToolkit パッケージの活用
 - Dynamo ポリラインをソリッドに変換
 - ・ Python スクリプトの活用



解説 7) パッケージの活用

パッケージの概要 & 使用方法

- パッケージの概要

- Autodesk 以外の第三者が開発したノードをまとめて配布したもの
- いわば“便利なノード集”

- 使用方法

- パッケージを検索する
- パッケージをダウンロードする
- ライブラリ内に、パッケージが追加される

- Civil 3D Toolkit パッケージ

- Civil 3D, AutoCAD ノードを大幅に拡張



インストール



管理・アンインストール

解説 7) パッケージの活用

Civil 3D Toolkit パッケージで出来るようになること（多すぎるので一部のみ抜粋）

- オブジェクトの作成
 - Arc.ByGeometry
 - ・ Arc を Dynamo ジオメトリに変換
 - PolylineExtensions.ObjectByGeometry
 - ・ Arc を含む Polyline を Dynamo ジオメトリに変換
 - BlockReferenceExtension.ByGeometry
 - ・ Dynamo ジオメトリから、ブロック参照を作成
 - BlockReferenceExtension.GetGeometry
 - ・ ブロック参照から、Dynamo ジオメトリを取得
 - AlignmentExtensions.CreateAlignmentByPolyline
 - ・ ポリラインから平面線形を作成
 - AlignmentExtensions.AddProfileByName / ProfileExtensions.AddPVI
 - ・ 空の縦断線形を作成 → PVI 追加
- オブジェクトの選択
 - Select Objects
 - ・ 複数のオブジェクトを一括で選択
 - Object Types
 - ・ Arc, CogoPoint, 計画線, ハッチング など、フィルタできるオブジェクトの種類が増加
 - Surface Names / Alignment Names / Corridor Names
 - ・ 名前を指定して、サーフェス, 線形, コリドー を取得
- オブジェクトから情報を取得
 - ObjectExtensions.SetParameterValueByName
 - ・ Civil 3D を含むオブジェクトに、パラメータを設定
 - ObjectExtensions.GetParameters
 - ・ Civil 3D を含むオブジェクトの、パラメータを取得

解説 7) パッケージの活用

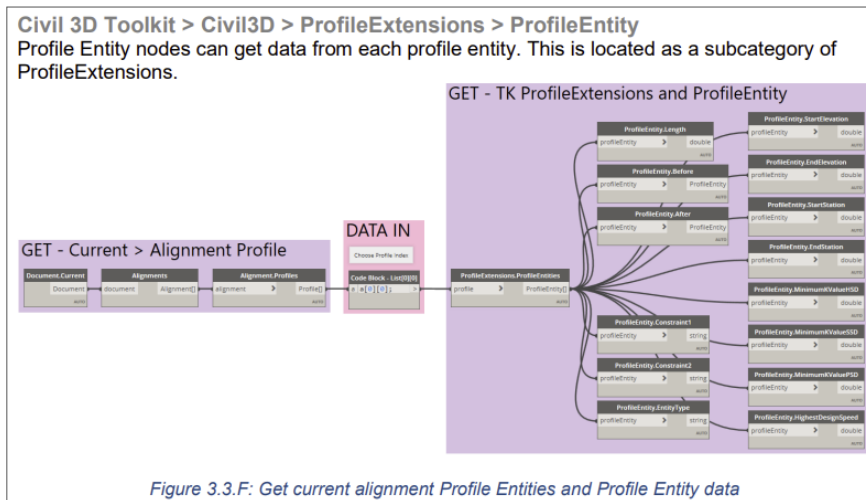
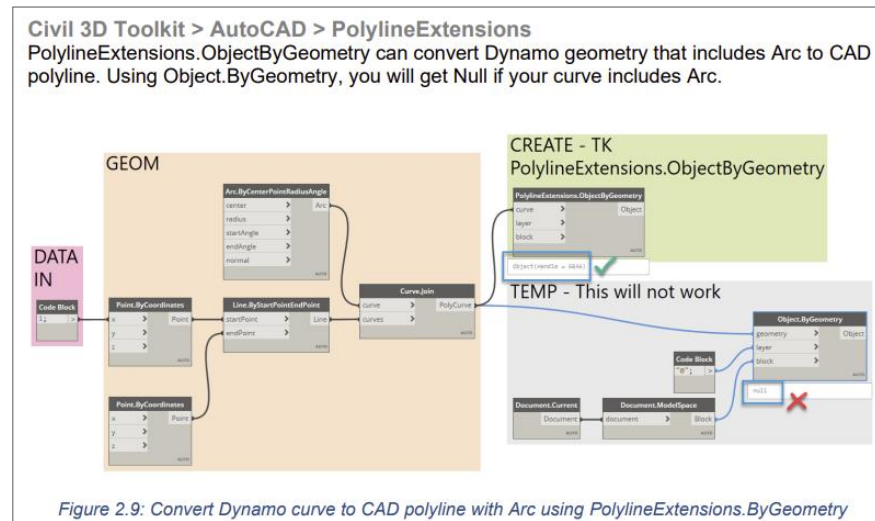
Civil 3D Toolkit パッケージで出来るようになること（多すぎるので一部のみ抜粋）

- 主な拡張項目 (AutoCAD)
 - ハッチング (Hatch)
 - テキスト、引出線 (MText, MLeader)
 - 凡例 (Legend, LegendItem)
 - 画層 (Layer)
 - ブロック (Block, BlockReference)
 - 表 (Table)
 - レイアウト (Layout, Viewport)
 - プロットスタイル (PlotStyle)
- 主な拡張項目 (Civil 3D)
 - サーフェス (TinSurface)
 - 線形 (Alignment)
 - 縦断 (Profile, PVI)
 - 縦断ビュー (ProfileView)
 - アセンブリ (Assembly, AssemblyGroup)
 - サブアセンブリ (Subassembly, ...Link, ...Shape, ...Point, ...TargetInfo, TkSubassemblyParameter)
 - コリドー (Corridor)
 - コリドー計画線 (TkCorridorFeatureLine)
 - 横断 (Section)
 - 横断ビュー (SectionView)

解説 7) パッケージの活用

Civil 3D Toolkit パッケージで出来るようになること（多すぎるので一部のみ抜粋）

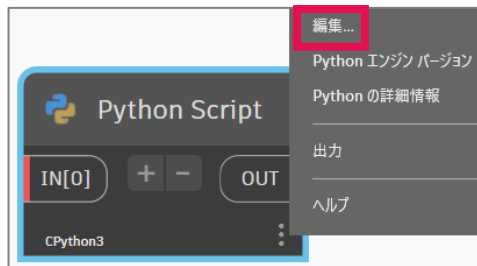
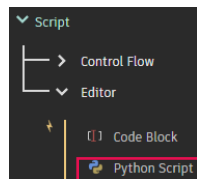
- Autodesk University セッションの中で、各ノードの機能を網羅的に解説
 - [Supercharge Your Dynamo Graph with Civil 3D Toolkit | Autodesk University](#)



解説 8) Python スクリプトの活用

Python Script テンプレートに記載されている内容は...

Python で Civil 3D, AutoCAD のデータを直接操作する (上級者向け)



```
Python Script
1 # Load the Python Standard and DesignScript Libraries
2 import sys
3 import clr
4
5 # Add Assemblies for AutoCAD
6 clr.AddReference('AcMgd')
7 clr.AddReference('AcCoreMgd')
8 clr.AddReference('AcDbMgd')
9
10 # Import references from AutoCAD
11 from Autodesk.AutoCAD.ApplicationServices import *
12 from Autodesk.AutoCAD.DatabaseServices import *
13 from Autodesk.AutoCAD.Runtime import *
14 from Autodesk.AutoCAD.EditorInput import *
15 from Autodesk.AutoCAD.Geometry import *
16
17 # Add Assemblies for Civil3D
18 clr.AddReference('AecBaseMgd')
19 clr.AddReference('AecPropDataMgd')
20 clr.AddReference('AeccDbMgd')
21
22 # Import references from Civil3D
23 from Autodesk.Civil.ApplicationServices import *
24 from Autodesk.Civil.DatabaseServices import *
25
26 # The inputs to this node will be stored as a list in the IN variables.
27 adoc = Application.DocumentManager.MdiActiveDocument
28 editor = adoc.Editor
29
30 with adoc.LockDocument():
31     with adoc.Database as db:
32         with db.TransactionManager.StartTransaction() as t:
33
34             # Here you can write your code...
35
36         t.Commit()
37
38 # Assign your output to the OUT variable.
39 OUT = 0
```

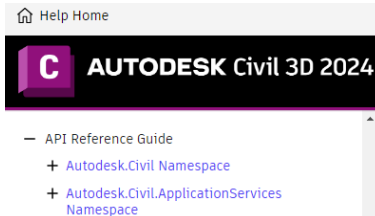
AutoCAD API にアクセスする
ための“おまじない”

Civil 3D API にアクセスする
ための“おまじない”

dwg 編集の開始&終了を宣言する
ための“おまじない”

プログラムを実装

* API Reference を参照しながら



解説 8) Python スクリプトの活用

今回のサンプルで作成した内容は...

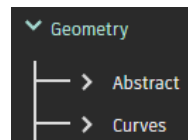
Python で繰り返し計算を行う、Python で Dynamo ジオメトリを操作する (中級者向け)

```
Python Script
# おまじない
import sys
import clr

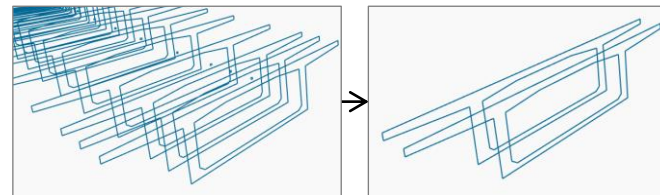
# Geometry フォルダのノードを扱う場合
clr.AddReference('ProtoGeometry')
from Autodesk.DesignScript.Geometry import *

9 input = IN[0]
10 diffSolidList = []
11
12 # Python の関数で繰り返し計算
13 for i in range(1, len(input)-1):
14
15     # ソリッドの開始断面と終了断面で、ポリラインを取得
16     outerPolyline = [input[i][0], input[i+1][0]]
17     innerPolyline = [input[i][1], input[i+1][1]]
18
19     # ポリラインから、ソリッドを作成
20     outerSolid = Solid.ByLoft(outerPolyline)
21     innerSolid = Solid.ByLoft(innerPolyline)
22     diffSolid = Solid.Difference(outerSolid, innerSolid)
23     diffSolidList.append(diffSolid)
24
25     # 要らないソリッドを消去
26     outerSolid.Dispose()
27     innerSolid.Dispose()
28
29 OUT = diffSolidList
30
```

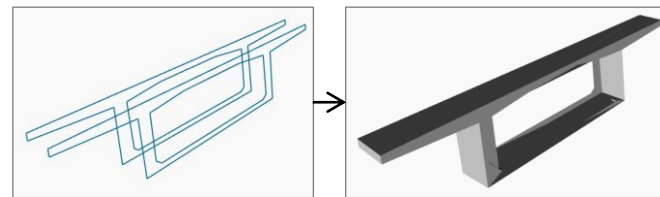
Dynamo の Geometry ライブラリに
アクセスするための“おまじない”



ソリッドの作成に必要な
ポリラインを抽出



ポリラインから
ソリッドを作成



使用しない図形を削除 (処理の高速化のため)

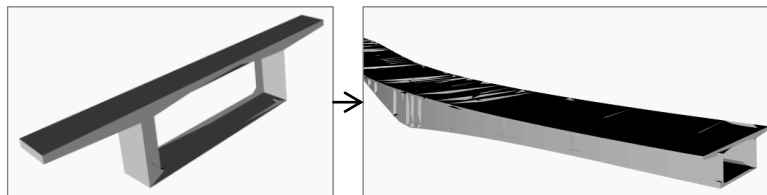
解説 8) Python スクリプトの活用

今回のサンプルで作成した内容は...

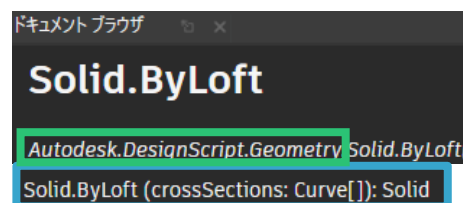
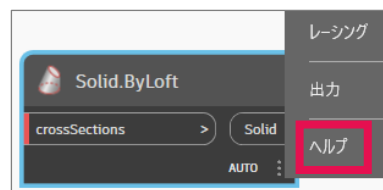
Python で繰り返し計算を行う、Python で Dynamo ジオメトリを操作する (中級者向け)

```
Python Script
1 | おまじない
2 | import sys
3 | import clr
4 |
5 | # Geometry フォルダのノードを扱う場合
6 | clr.AddReference('ProtoGeometry')
7 | from Autodesk.DesignScript.Geometry import *
8 |
9 | input = IN[0]
10 | diffSolidList = []
11 |
12 | # Python の関数で繰り返し計算
13 | for i in range(1, len(input)-1):
14 |     # ソリッドの端部断面と終了断面で、ポリラインを取得
15 |     outerPolyline = [input[i][0], input[i+1][0]]
16 |     innerPolyline = [input[i][1], input[i+1][1]]
17 |
18 |     # ポリラインから、ソリッドを作成
19 |     outerSolid = Solid.ByLoft(outerPolyline)
20 |     innerSolid = Solid.ByLoft(innerPolyline)
21 |     diffSolid = Solid.Difference(outerSolid, innerSolid)
22 |     diffSolidList.append(diffSolid)
23 |
24 |     # 要らないソリッドを消去
25 |     outerSolid.Dispose()
26 |     innerSolid.Dispose()
27 |
28 |
29 | OUT = diffSolidList
30 |
```

ソリッドを作成 の処理を
繰り返し実施



Dynamo ノードの
Python での記法は
ヘルプを見ながら



```
5 | # Geometry フォルダのノードを扱う場合
6 | clr.AddReference('ProtoGeometry')
7 | from Autodesk.DesignScript.Geometry import *
8 |
9 | # ポリラインから ソリッドを作成
10 |
11 | outerSolid = Solid.ByLoft(outerPolyline)
12 | innerSolid = Solid.ByLoft(innerPolyline)
```