

## 平成 28 年度 文部科学省委託事業実証講座

社会基盤分野における建設 IT 技術 (BIM・CIM) に係る  
中核的専門人材養成プログラム開発プロジェクト

### 【建築 BIM 講座】

—コンピューターショナル・デザインとは—

Vectorworks Architect で学ぶ  
マリオネット ビジュアルプログラミング

H28.12

A&A



ようこそ、マリオネットの世界へ

マリオネットは Vectorworks  
を使うデザイナーのためのビジ  
ュアルプログラミング環境です。

このガイドブックをきっかけに、  
ぜひ新しいデザインの世界を体  
験してください。

# マリオネット実践ガイド

# マリオネット実践ガイド

## 目 次

<b>第1章 ビジュアルプログラミングとは .....</b>	<b>1</b>
<b>なぜビジュアルプログラミング .....</b>	<b>2</b>
<b>第2章 マリオネットの基本.....</b>	<b>7</b>
1. マリオネットとは.....	8
2. マリオネットをはじめる .....	9
3. ノードのしくみ.....	11
4. マリオネットネットワークとデータフロー .....	18
5. マリオネットネットワークのデバッグ .....	22
<b>第3章 マリオネット実践 .....</b>	<b>23</b>
1. 球の構造物をつくる .....	24
2. 太陽光設定の情報を利用して南中高度を計算する .....	31
3. ワークシートから風配図をつくる .....	37
4. 光と風に配慮したパラメトリックなパネルをつくる .....	49
<b>マリオネット関連情報.....</b>	<b>59</b>
<b>お問い合わせ .....</b>	<b>59</b>

# **第1章 ビジュアルプログラミングとは**

## なぜビジュアルプログラミング

「デザイン」というと人の手から生まれるもので、コンピュータのプログラムなんて関係ないと思われるかもしれません、実はそんなことはないんです。たぶん、この文章を読んでいる皆さんも、そんなことを感じられて目を通されているのではないですか。

コンピュータのプログラムは物事を処理する順序を定めたルールと言えます。ルールをデザインの中に取り入れて、新たな形態や空間を発想するということは、黄金比や白銀比、木割やモジュールなどなど、挙げればきりがないほどで、非常に古くから行われています。

面積を変えずに形を変えたいとか、建物の生産プロセスに基づいて寸法を決めるなど、これらのルールの出自は様々ですが、なぜデザインの過程にルールが取り入れられるのでしょうか？

ひとつには、ルールは様々なデザインに適用できるからです。木の葉が枝の先に行くにしたがつて小さくなる様子は多くの樹木で共通しており、このような自然の法則、ルールをデザインに応用する例は数多くあります。一つのルールが、多くの場面で適用でき、様々なデザインを生み出すところも魅力の一つと言えそうです。

もう一つは、視点を変える面白さがあると思います。例えば、同じ四角形を描くというテーマでも、ルールを変えるだけで、多種多様なデザインが生まれます（図 1）。

中でも一番魅力的のは、一つのルールから様々なデザインを生み出せることです。ルールに与える条件を変えることで、同じルールでも異なる結果が得られるのです。そして、時にはその結果が予想もしていない素敵なかたちになります。

このルールに与える条件のことをパラメーターと呼ぶことがあります、このパラメーターを様々に変化させ、ルールにより多様な形を作り出す手法を、パラメトリックデザインと呼びます。

例えば線画で長方形をデザインするとしたら、パラメーターは、幅と高さでしょうか。点描でデザインするなら、点の密度ということになりそうです。このように、ルールに応じて何がパラメーターとなるかは変わりますが、一つのルールでも、パラメーターがどのような値をとるか（幅がいくつで高さがいくつ）によって、様々な形ができるのです。これがパラメトリックデザインの考え方です。

少し建築的な例としては、L字型プランの住戸を配置する例で、ルールとしては、4つの住戸を移動して並べることとして、その移動方法をパラメーターとしたものが図2です。このように、単純なルールとパラメーターですが、数多くのバリエーションが作られることがわかります。こうしたバリエーションを作り出すのがパラメトリックデザインの特徴です。

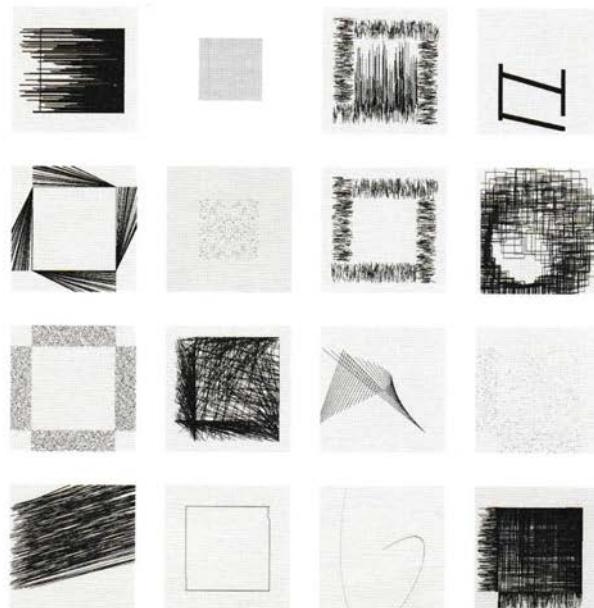


図1 様々なルールで描かれた四角形（出典：「デザイン・ウィズ・コンピュータ」永原康史著）

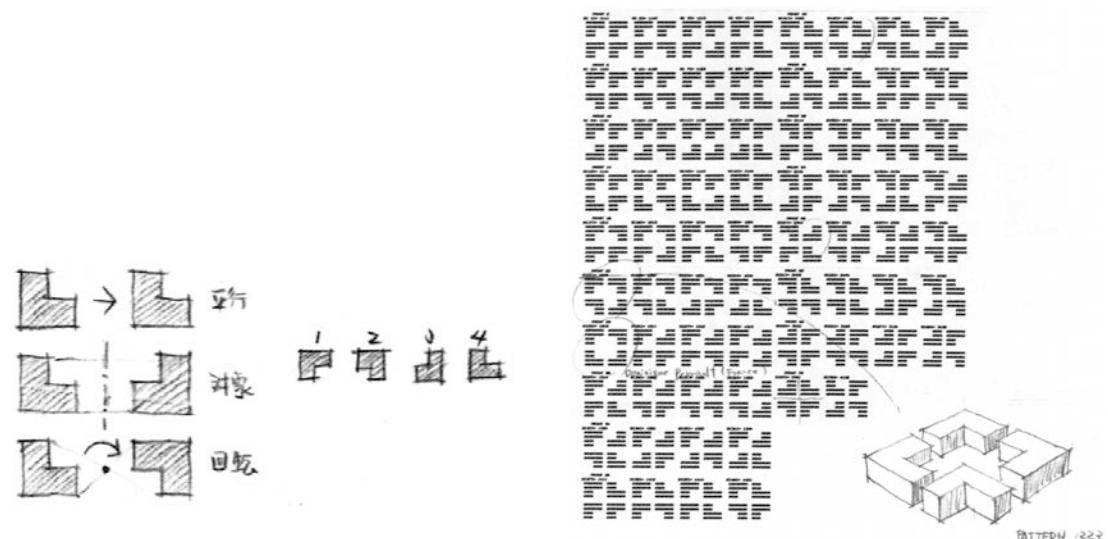


図2 住戸プランのパラメトリック配置（出展：「アルゴリズミック・デザイン」日本建築学会編）

黄金比で四角を描くというのもパラメトリックデザインの一つですが、この機能は、Vectorworks の標準ツールですでにできるようになっています。

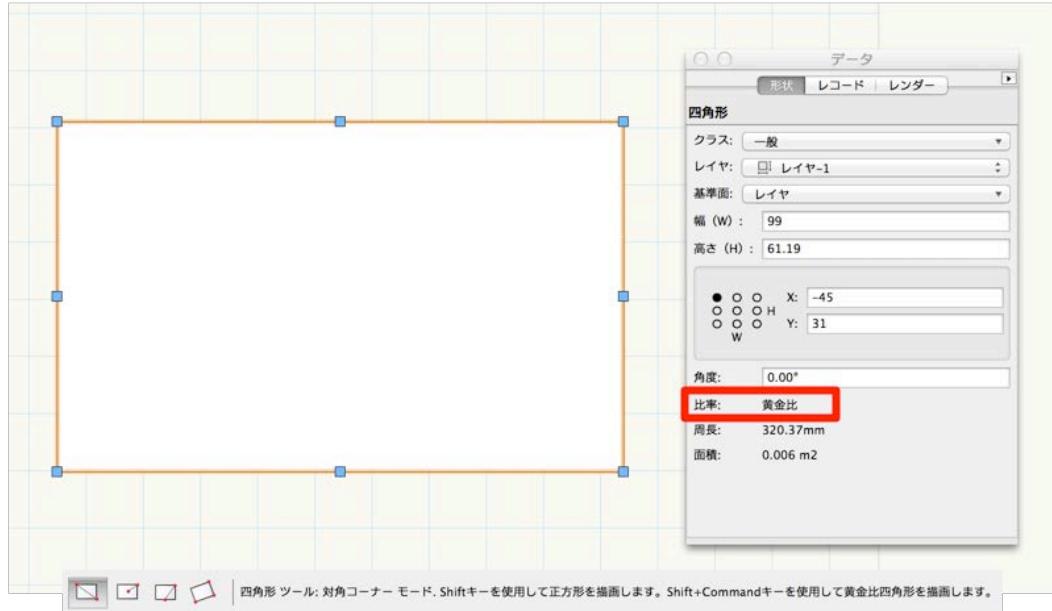


図 3 Vectorworks の四角形ツールで黄金比图形を作成

このように、CAD のソフト自体もパラメトリックデザインのツールと言えますが、もう少し凝ったことをしようと思うと、自分でプログラムを作る必要があります。

さて、プログラムというと、一般的には図 4 のように暗号のようなテキストのリストを思い浮かべられると思います。このプログラムは実際には 100 行くらいありますが、デザインをするつもりでいて、いきなりこれを間違いなく完成させるのはなかなか難しいでしょう。

```

スクリプトエディタ

スクリプト言語: Pythonスクリプト リンク元: スクリプト / スクリプト-1

1 import Marionette
2 import math
3 import operator
4 Marionette.execution_mode = Marionette.ExecutionType.Export
5 ##### Real_324 #####
6 class Real_324:
7     Handle = vs.Mrntte_GetNodeHandle(324)
8     @Marionette.NodeDefinition
9     class Params(metaclass = Marionette.OrderedClass):
10         this = Marionette.Node('Real')
11         this.SetDescription('A Real value defined in the OIP in document units')
12         r = Marionette.PortOut()
13         r.SetDescription('The value')
14         Real = Marionette.OIPControl('real', Marionette.WidgetType.Real, 0.0)
15         Real.SetDescription('A value in document units')
16
17     def RunNode(self):
18         self.Params.r.value = self.Params.Real.value

```

図 4 テキスト記述のプログラム

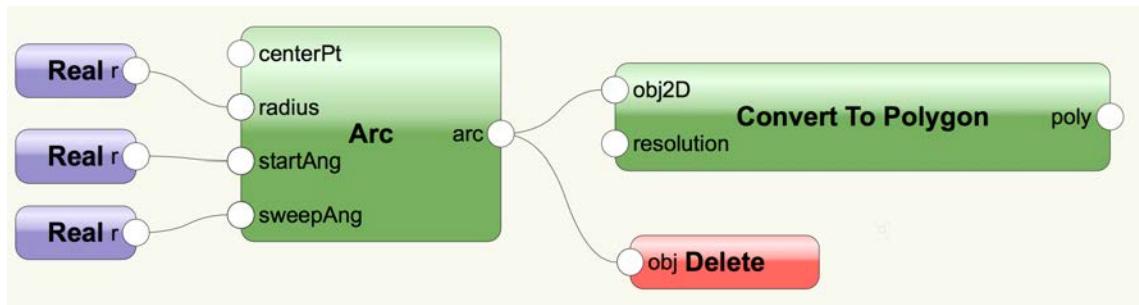


図 5 ビジュアルプログラミングの例

これに対して、ビジュアルプログラミングでは図 5 のようにビジュアルに図形を配置してプログラムを作ります。この図は円弧を描いて多角形に変換するプログラムですが、先ほどのテキストで 100 行くらいあるものと同じことをします。

CAD でデザインをする人にとってビジュアルプログラミングは馴染みやすく、また、作ったプログラムをシェアしてそこからオリジナルの形を作るなど、デザインの現場で使われることが多くなり、デザインの可能性を広げる、新しい発見を得るチャンスとして注目されています。

ところで、Vectorworks には数多くの作図機能が用意されていますが、それらを補う形でビジュアルプログラミングを利用することができます。

例えば、図形を配置する際に、配列複製やパス複製の機能で等間隔に図形を並べることはできますが、適度にばらつかせて配置をしようと思うと、 ランダムノードの出番です。

他にも、らせん状に図形を配置したり、少しずつ大きさを変えて配置するなどと、少し複雑に配置したいときにもプログラミングが役に立ちます（図 7）。

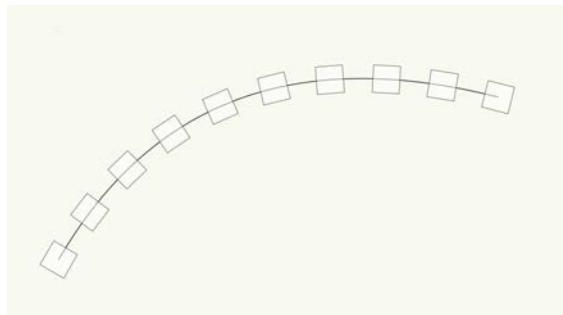


図 6 「パス複製」による図形の等間隔配置

このように、Vectorworks の機能から発想をふくらませてパラメトリックデザインを展開するともできそうです。

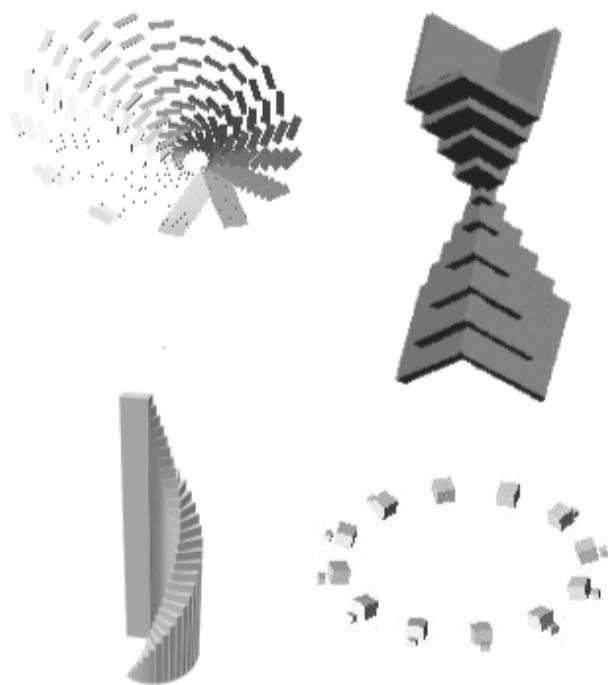


図 7 ちょっと複雑な配置（出展：「アルゴリズミック・デザイン」日本建築学会編）

パラメトリックデザインは、ルールやパラメーターが複雑になるとアルゴリズミックデザインと呼ばれたり、もう少し幅を広げてコンピュテーションナルデザインと呼ばれ発展しています。何れにしても、プログラムと協同して多様なバリエーションを作つて、良いデザインを見つけ出すというところは変わりません。この機会に、ぜひビジュアルプログラミングに親しみ、新たな可能性にチャレンジしてください。

【参考文献】アルゴリズミック・デザイン 日本建築学会編 鹿島出版会

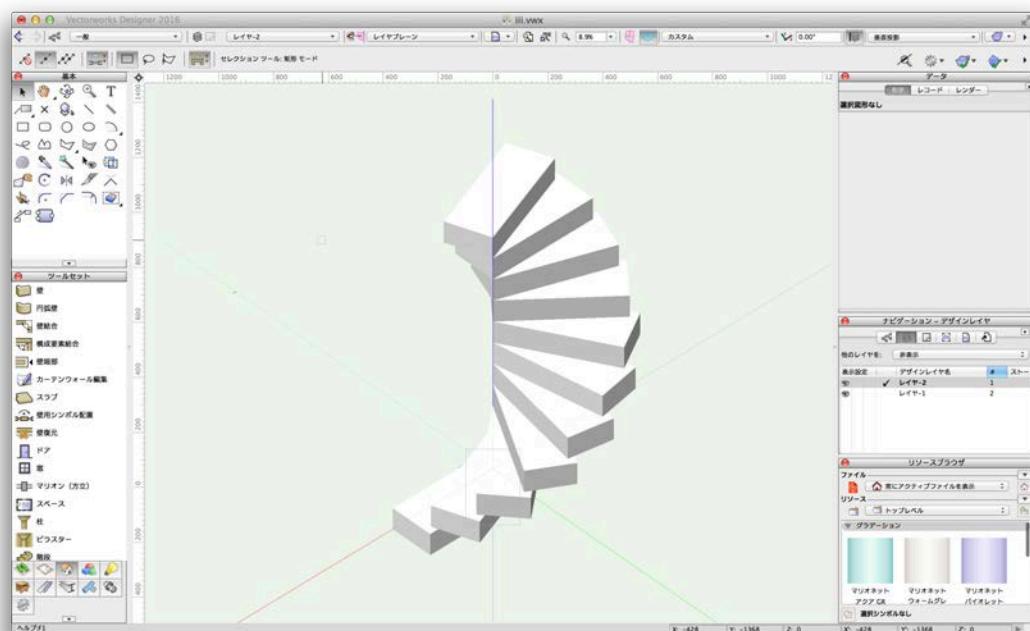
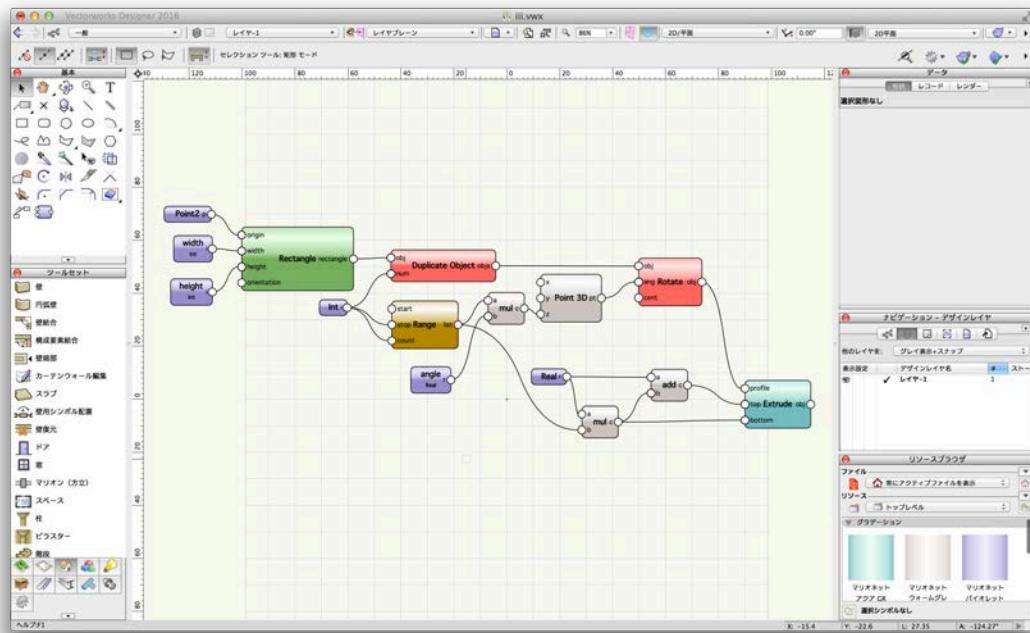


## **第2章 マリオネットの基本**

# 1. マリオネットとは

マリオネットはVectorworksに組み込まれた、ビジュアルプログラミング環境です。たったひとつの「マリオネットツール」からその世界は始まります。

コンピュテーションナルデザインやパラメトリックデザインなど、プログラミングで様々なデザインの可能性が開けます。そんなプログラミングが、ぱちぱちテキストを打ち込むことなくできてしまいます。「ノード」と「ワイヤー」という図形をつなげるだけでプログラミングができるのです。



## 2. マリオネットをはじめる

### 2.1. ノードを選択する

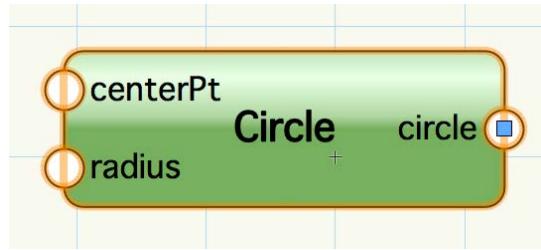
マリオネットツールを使って、Vectorworks にあらかじめ用意されたノードを使用することができます。

マリオネットツールには、図形を描画するもの、図形の属性を編集するもの、数学の演算を行うもの、データの流れを管理するものなど、500 種以上のノードが用意されています。



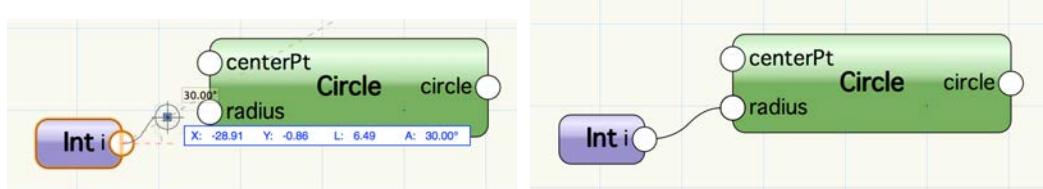
### 2.2. ノードを図面に配置する

マリオネットツールがアクティブになっていると、選択したノードのシルエットが図面上のカーソルに追従し、クリックした位置に、選択したノードが配置されます。



### 2.3. ノードを繋いでマリオネットワークを作成する

ノードの出力ポートからワイヤーを伸ばして他のノードの入力ポート上でクリックすることで、ノードとノードを繋ぐことができます（※ビューを2D/平面に設定しておく必要があります）。ワイヤーで繋がれたノードの集合をマリオネットワークと呼びます。

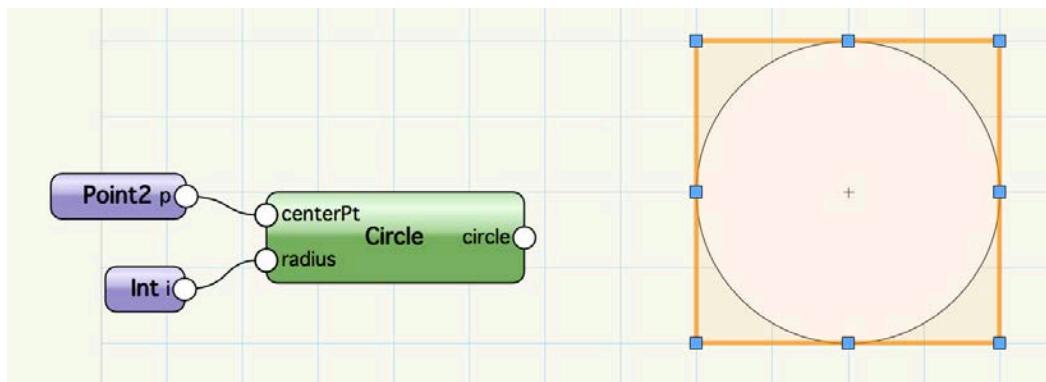


### 2.4. マリオネットスクリプトの実行

作成したマリオネットワークは一連の動作をするスクリプトになっており、次のいずれかの操作で実行することができます。

- ・ ノードを右クリック > コンテキストメニュー > マリオネットスクリプトを実行
- ・ ノードを選択 > データパレット > 実行ボタン

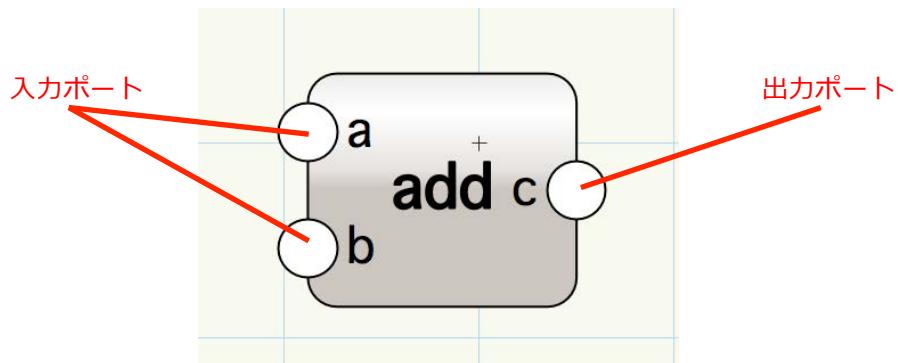
スクリプトの実行により新たに作成されたオブジェクトはグループ化されます。



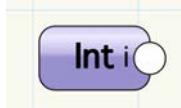
### 3. ノードのしくみ

#### 3.1. ノードとは

ノードは、マリオネットの部品のひとつで、それぞれにスクリプトでコマンドが定義されています。ノードは、コマンドの実行に必要なデータを左側にある入力ポートから受け取り、定義されたコマンドを実行して、右側にある出力ポートからその結果を出力します。



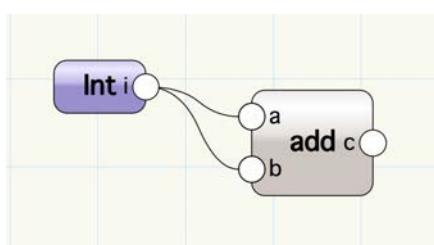
出力のみのノード：データを受け取らず、結果の出力のみ行います。



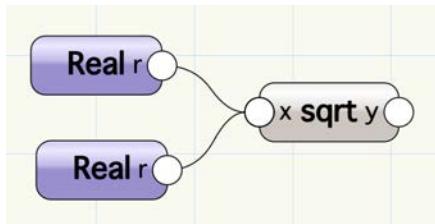
入力のみのノード：データを受け取り、コマンドを実行しますが結果は出力しません。



ノードはひとつの出力ポートから複数のワイヤーを伸ばすことができます。

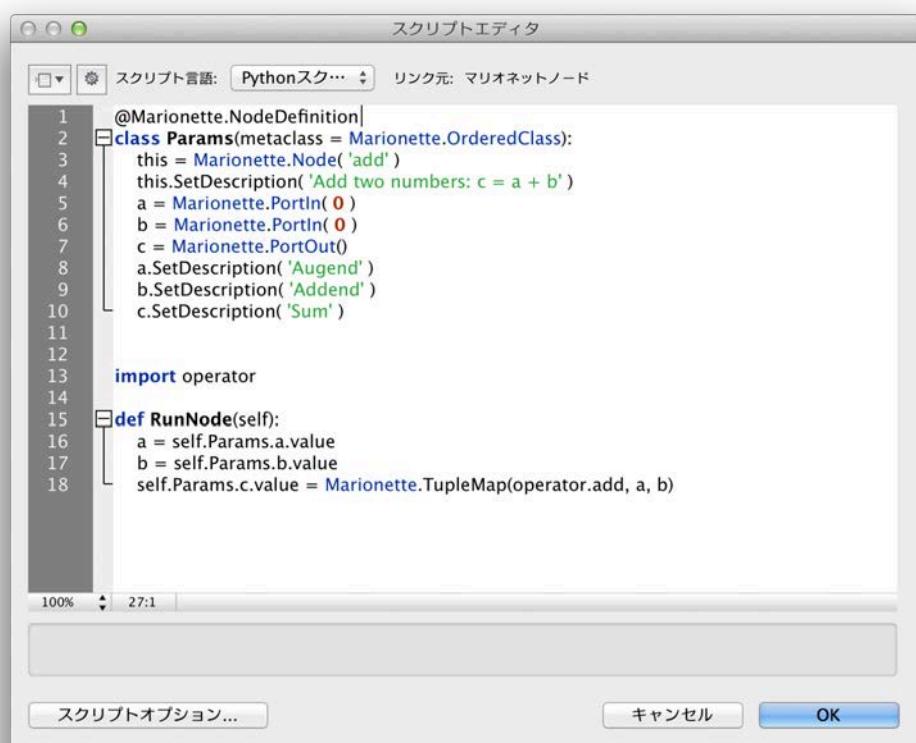


ひとつの入力ポートに複数のワイヤーを接続することができます。



### 3.2. ノードのコマンドの編集

ノードのコマンドを定義するスクリプトは Python で記述されており、スクリプトエディタで編集することができます。Python の知識があれば、既存のノードを編集することで、オリジナルのノードを作成することができます。

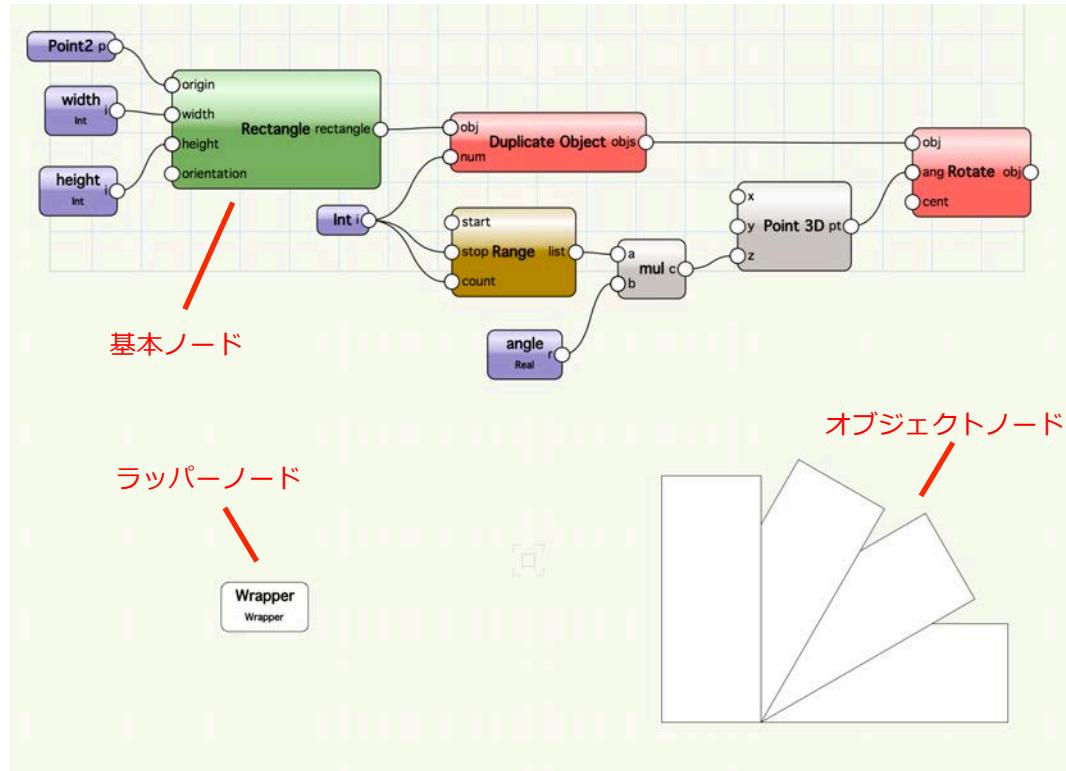


#### 【スクリプトエディタの開き方(3 パターン)】

- ・ ノードを選択 > データパレット > 編集ボタン
- ・ ノードを右クリック > コンテキストメニュー > スクリプトの編集
- ・ ノードをダブルクリックする

### 3.4. ノードの種類

マリオネットのノードは、3種の型（基本ノード・ラッパーノード・オブジェクトノード）があります。

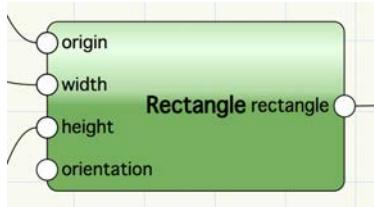


それぞれのデータパレットの表示に違いがあります。



## 1) 基本ノード

ノードの基本の型であり、コマンドのスクリプトを直接内包しています。



【基本ノードのデータパレット】

- ・ 編集ボタン：コマンドを定義しているスクリプトを編集するエディタを開く。
- ・ 説明ボタン：コマンドの動作の概要を表示する。
- ・ 実行ボタン：ノードが接続されているマリオネットワークのスクリプトを実行する。
- ・ 名前：選択中のノードに名前を設定する。
- ・ タイプ：ノード種類を示す。
- ・ パラメータ：ノードの動作に影響する値を入力する。

## 2) ラッパーノード

複数のノードを繋いだネットワークを、単一のノードに統合したノードです。マリオネットネットワークの整理に用います。



【ラッパーノードの作成手順】

ネットワーク右クリック > コンテキストメニュー > マリオネットネットワークをラップ

【ラッパーノードの作成手順】

ネットワーク右クリック > コンテキストメニュー > マリオネットネットワークをラップ

【ラッパーノードの解除手順】

ラッパーノード右クリック > コンテキストメニュー > マリオネットネットワークのラップを解除

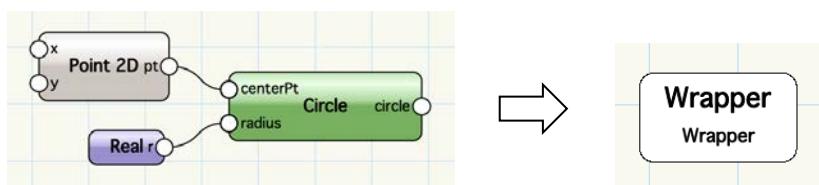
### 【ラッパーノードのデータパレット】

- ・ 編集ボタン：内包しているマリオネットネットワークの編集画面を開く。
- ・ 実行ボタン：ノードが接続されているマリオネットネットワークのスクリプトを実行する。
- ・ 名前：選択中のラッパーノードに名前をつける。
- ・ パラメータ：内包するノードのうち名前をつけたノードのパラメータが表示される。
- ・

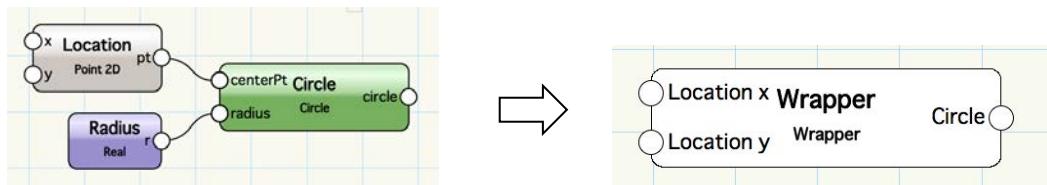
### 【ラッパーノードに入出力ポートを表示する】

マリオネットネットワークをラッパーノードに変換する際、各ノードに名前を設定しておくと、名前をつけたノードの未接続の入出力ポートはラッパーノードの入出力ポートとして、パラメータはラッパーノードのデータパレットにフィールドとして表示されます。

#### “ノードに名前をつけない場合”



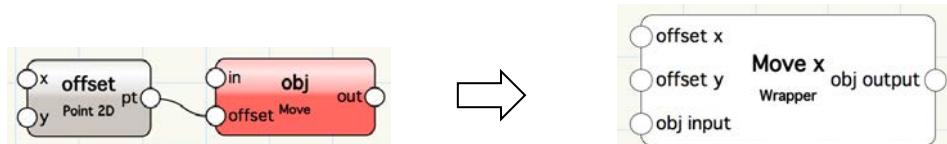
#### “ノードに名前をつけた場合”



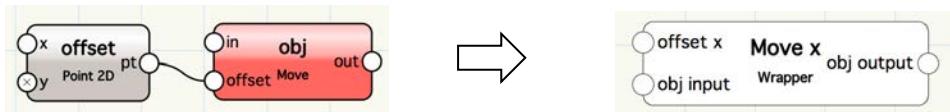
### 【未接続のポートを非表示にする】

名前をつけたノードの未接続の入出力ポートは、ポートの位置に2D基準点を配置すると、ラッパーノードでは非表示になります。

#### “未接続のポートに2D基準点を配置しない場合”

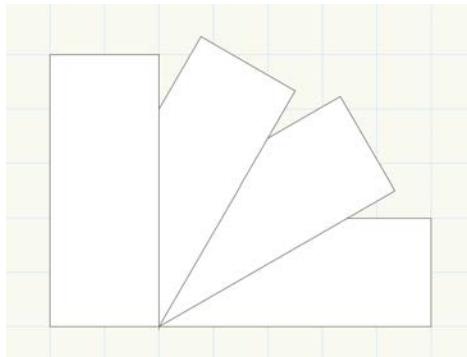


#### “未接続のポート(yポート)に2D基準点を配置した場合”



### 3) オブジェクトノード

ラッパーノードと同じく、マリオネットネットワークをひとつに統合したノードです。オブジェクトノードは内包するマリオネットスクリプトの実行結果の図形として表示され、スクリプトの実行を必要としません。ネットワークの変更やパラメータの変更がすぐに図形の表示に反映されます。



#### 【オブジェクトノードの作成手順】

ラッパーノードを右クリック > コンテキストメニュー > オブジェクトノードに変換

#### 【オブジェクトノードの解除手順】

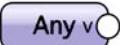
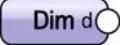
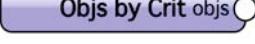
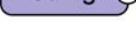
オブジェクトノードを右クリック > コンテキストメニュー > ラッパーノードに変換

#### 【オブジェクトノードのデータパレット】

パラメータ：内包するノードのうち名前をつけたノードのパラメータが表示される

### 3.5. 入力関数ノード

マリオネットネットでは、数値、座標、真偽、図形など様々な型のデータを扱うことができます。マリオネットツールにはこれらのデータを入力データとしてネットワークに接続するノードが用意されています。

	Python の eval 関数を使用して任意の値を作成する。
	データパレットのチェックボックスから、True または False の値を生成する。
	マリオネットオブジェクトの定義ネットワークで使われる時、コントロールジョメトリグループの最初のオブジェクトを返す。オブジェクトの外で使用できない。
	寸法としての実数を設定する。
	整数値を返す。
	名前の付いた図形を図面上から取得する。
	指定した検索条件に合致するオブジェクトのリスト返す。
	データパレットで定義した 2D 座標を返す。
	データパレットで定義した 3D 座標を返す。
	データパレットから定義する実数値を返す。ファイルの単位が反映されている。
	データパレットで定義する文字列を返す。
	2 次元ベクトルを作成する。
	3 次元ベクトルを作成する。

## 4. マリオネットワークのデータフロー

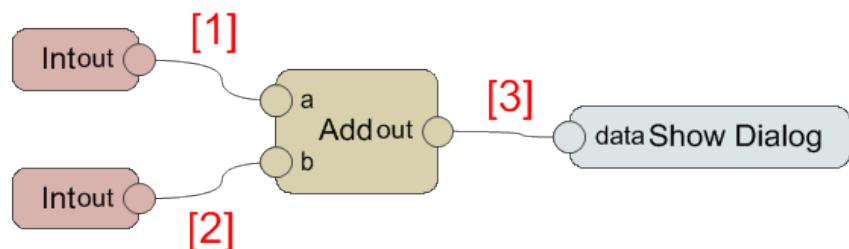
マリオネットワークは相互にデータのやりとりをするノードが集まつたものであり、各ノードで定義されたコマンドを順番に実行していきます。実行の順番はデータフローに依存し、データは一方向にしか流れません。データフローは、各ノードの入出力ポートを繋ぐワイヤーにより定義されます。

### 4.1. データフロー

ネットワークに接続されたノードは、入力ポートが受け取る値のタイプによって 4 パターンの方法で処理を行います。

#### 1) 単一の値を受け取る場合

それぞれのポートで受け取った単一の値を用いて、1 度だけコマンドを実行します。



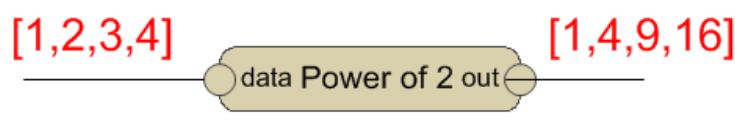
赤い文字はワイヤーを通過する値を示します。

【実行されるコマンド】

$$1 + 2 = 3$$

#### 2) 値のリストを受け取る場合

ノードは複数の値を出力する場合があります。このデータは値のリストとし入力ポートに渡されます。一般的なノードはリストを受け取ると、リストの要素をひとつずつ順番に取得しながら、要素の数だけコマンドを繰り返し実行します。



【実行されるコマンド】

$$1 * 1 = 1$$

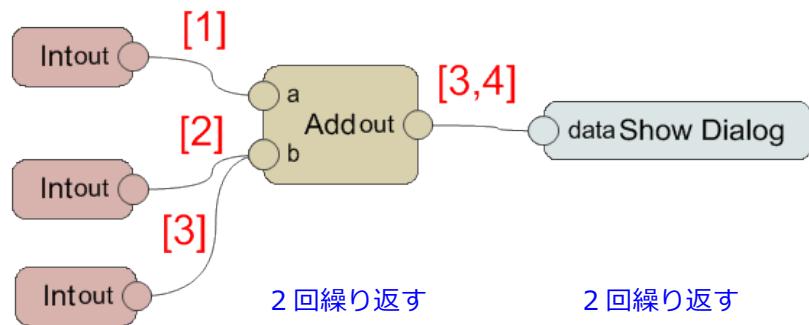
$$2 * 2 = 4$$

$$3 * 3 = 9$$

$$4 * 4 = 16$$

### 3) 要素数の違うリストが混合する場合

複数の入力ポートを持つノードで、各ポートに要素数の違うリストを受け取ると、通常は最も長いリストの要素数だけコマンドが実行されます。入力されるリストがすべて同じ長さになるように、要素数が少ないリストは、最後の値で補間されます。



#### 【使用するデータ】

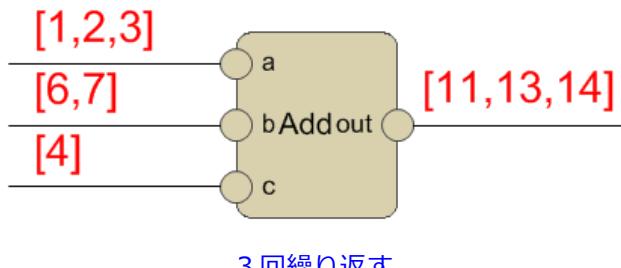
a ポート : [ 1, 1 ]

b ポート : [ 2, 3 ]

#### 【実行されるコマンド】

$$1 + 2 = 3$$

$$1 + 3 = 4$$



#### 【使用するデータ】

a ポート : [ 1, 2, 3 ]

b ポート : [ 6, 7, 7 ]

c ポート : [ 4, 4, 4 ]

#### 【実行されるコマンド】

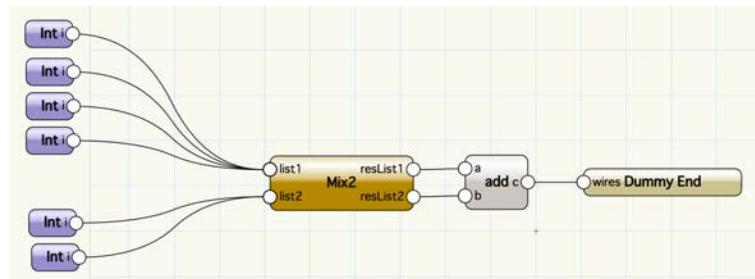
$$1 + 6 + 4 = 11$$

$$2 + 7 + 4 = 13$$

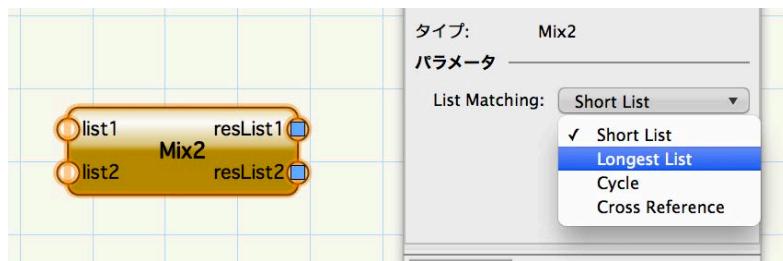
$$3 + 7 + 4 = 14$$

## 4.2. 長さの違うリストを任意の方式で揃える

長さの違うリストを目的のノードに渡す前に、Mix2 ノード(データフローカテゴリ)を通過させることで、リストの長さの揃え方を 4 パターンから選ぶことができます。



揃える方法は Mix2 ノードの List Matching パラメータから選択します。



### 【 Short List 】

最も短いリストの要素数に揃える。長いリストではみ出す値は切り捨てられます。



### 【 Longest List 】

最も長いリストの要素数に揃える。短いリストは最後の値で補間します。通常のノードはこの処理を行います。



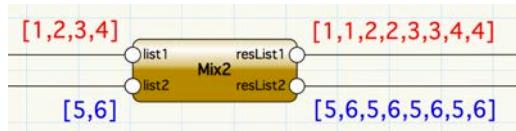
### 【 Cycle 】

最も長いリストの要素数に揃えます。短いリストは要素数を満たすまで最初の値から順に補間します。



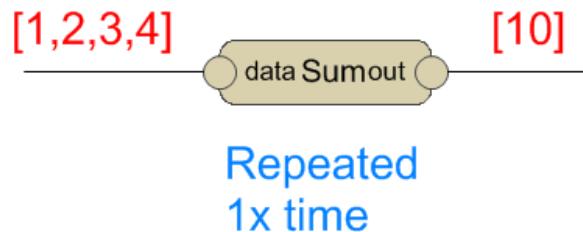
### 【 Cross Reference 】

要素数に関係なく、すべての値の組み合わせを取得できるリストを作成します。



### 4.3. リストを一括で受け取るノードの処理

ノードが List Absorb として定義されている場合、受け取ったデータはすべてひとつのリストに連結されます。このノードではリスト自体をひとつの値として取得します。そのため、受け取るデータのタイプに関わらずコマンドは 1 度だけ実行されます。

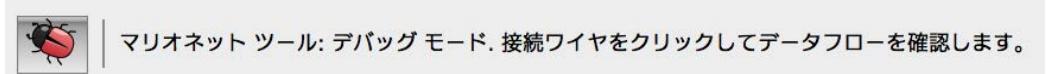


#### 【実行されるコマンド】

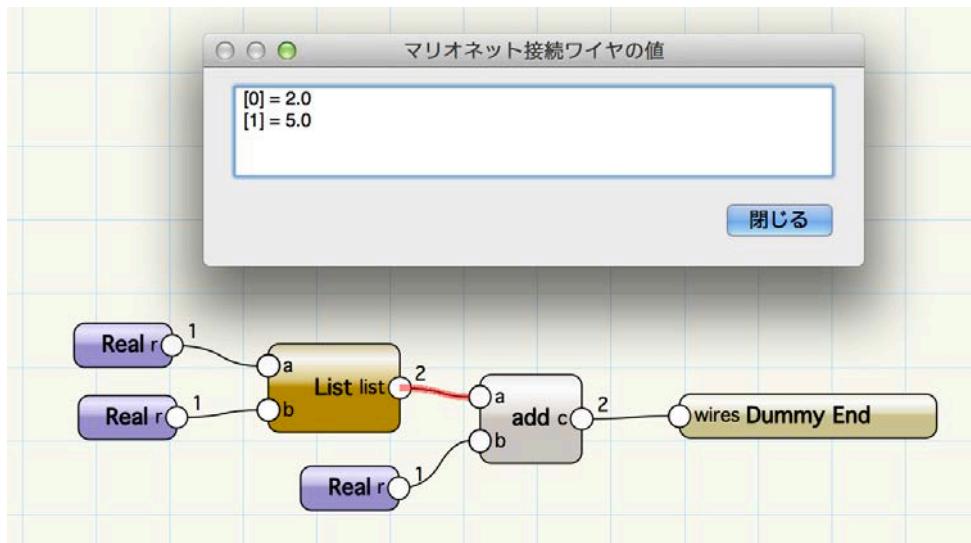
$$1 + 2 + 3 + 4 = 10$$

## 5. マリオネットワークのデバッグ

マリオネットツールのデバッグモードを用いて、ネットワークのトラブルシューティングが行なうことができます。



デバッグモードでは、各出力ポートの隣に数字が表示され、その出力ポートから接続された入力ポートに流れる値の数が表示されます。ワイヤーを選択するとワイヤーを通過する値がダイアログに表示され、ダイアログ内で値の確認と変更を行うことができます。ワイヤーを通過する値がハンドル型の場合、図形のタイプ番号が表示されます。ダイアログを閉じるとスクリプトが実行されます。



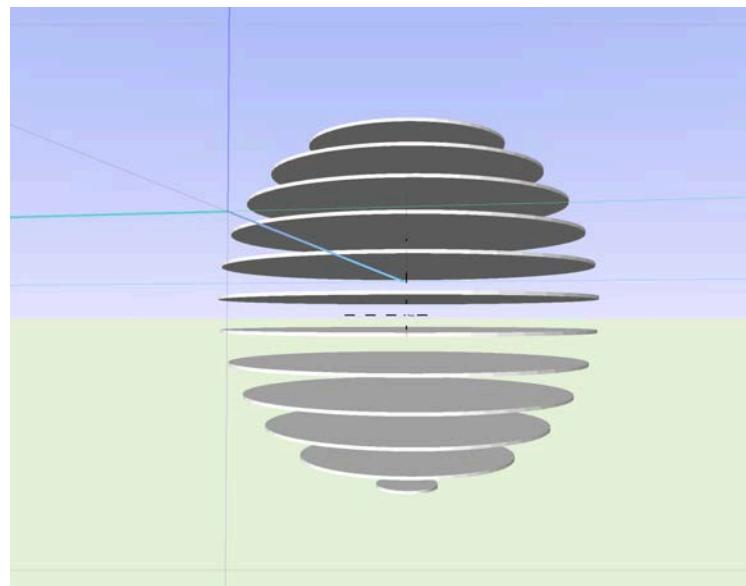
## **第3章 マリオネット実践**

# 1. 球の構造物をつくる

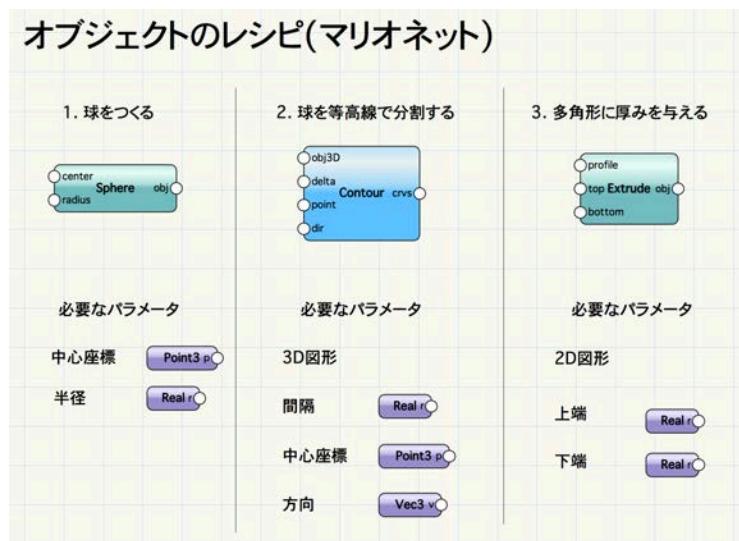
## 1-1. 球をひとつつくる

マリオネットで球を加工して次のような構造物を作成します。

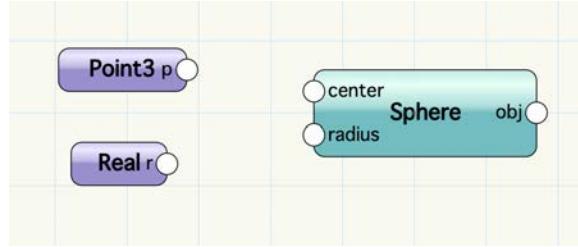
- [1] 球を作図する
- [2] 球を等高線で輪切りにする
- [3] 輪切りにした図形に厚みを与える



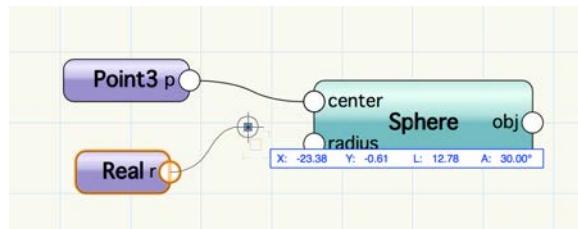
- 1) マリオネットで作図を行うためにマリオネットツールのノード一覧から各手順に必要なノードを選択します。各手順の処理を Vectorworks に命令するため次のようなノードを用意します。



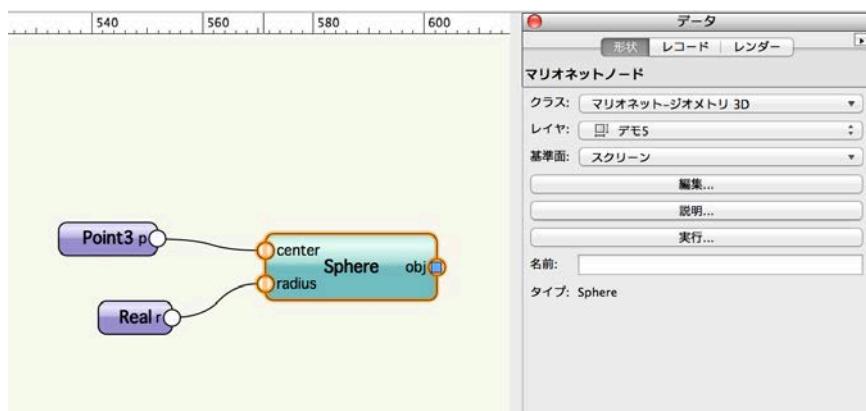
- 2) まず手順 1 の球をつくります。球を作図する「Sphere」ノードに加えて球を「どこに」「どの大きさの半径で」で作図するかを入力するためのノードを用意します。「どこに」に対して座標を入力する「Point3」ノードと、「どの大きさで」に対して実数値を入力する「Real」ノードを配置します。



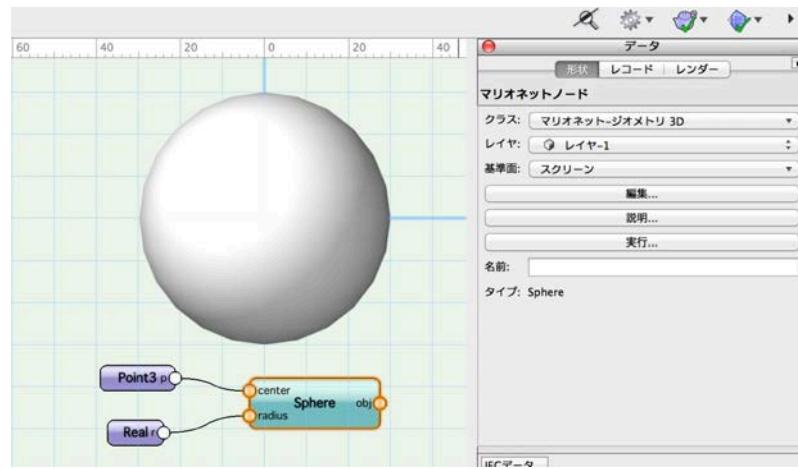
- 3) 必要なノードを配置したら入力用のノードを選択してデータパレットから値を設定します。ワイヤーで接続してネットワークを作成します。入力用ノードの出力ポート上でクリックしてワイヤーをのばし、「Sphere」ノードの入力ポート上でクリックします。



- 4) ネットワークのプログラムをデータパレットの実行ボタンから実行します。

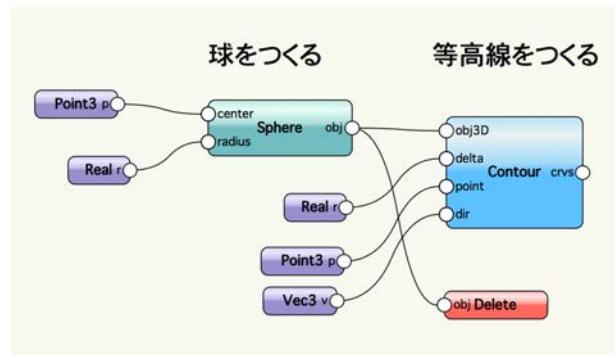


5) 図面上に球が作図されます。

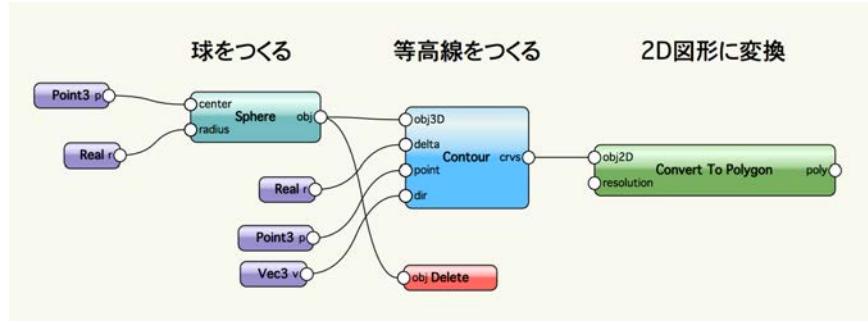


## 1-2. ネットワークをつくる

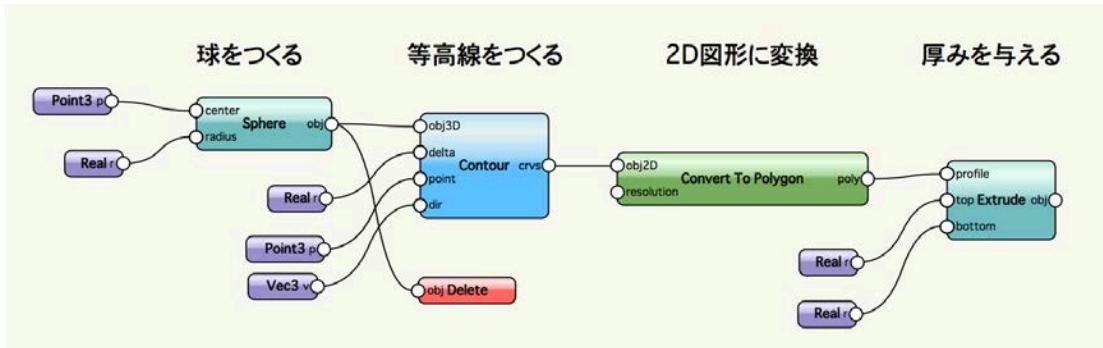
- 1) 「Sphere」ノードから出力される手順 1 の実行結果(球图形)を、手順 2 の等高線を作成する「Contour」ノードに入力します。“delta”に等高線の間隔、“point”に分割の開始位置の座標、“dir”に分割する方向のベクトル値を入力します。  
また、「Sphere」で作成した球は「Contour」で等高線を作図したあと不要となるため「Delete」ノードに入力して削除します。



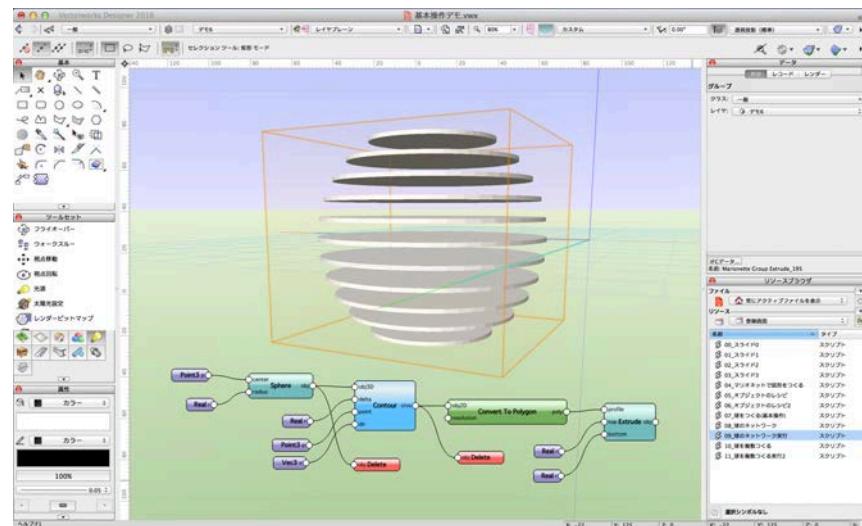
- 2) 手順 2 の実行結果の等高線を「Convert to Polygon」ノードをつかって 2D の多角形图形に変換します。これは、次の手順で接続するノードが受け付ける型に対応させるために必要な処理です。



- 3) 最後に「Extrude」ノードをつかって等高線に厚みを与えます。「Extrude」ノードには"top - bottom"が"厚み"となるように「Real」ノードをつかって値を入力します。

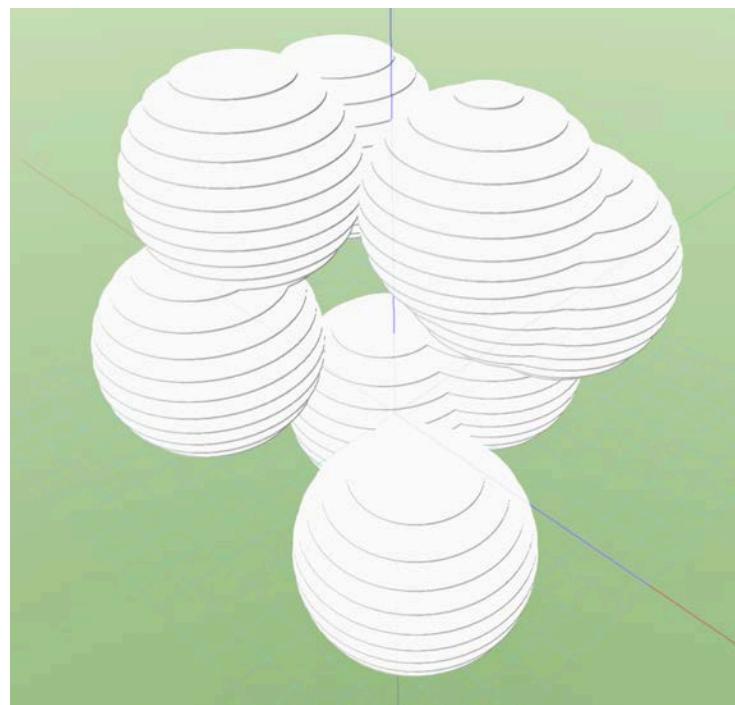


- 4) データパレットの実行ボタンを選択すると、各手順が順番に実行され球状の構造物が作図されます

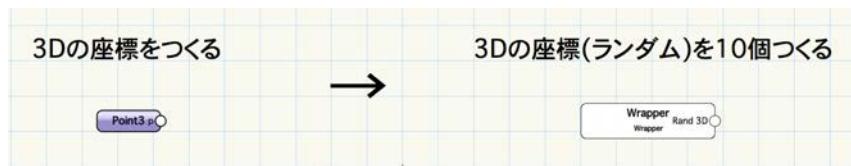


### 1-3. 複数の球をつくる

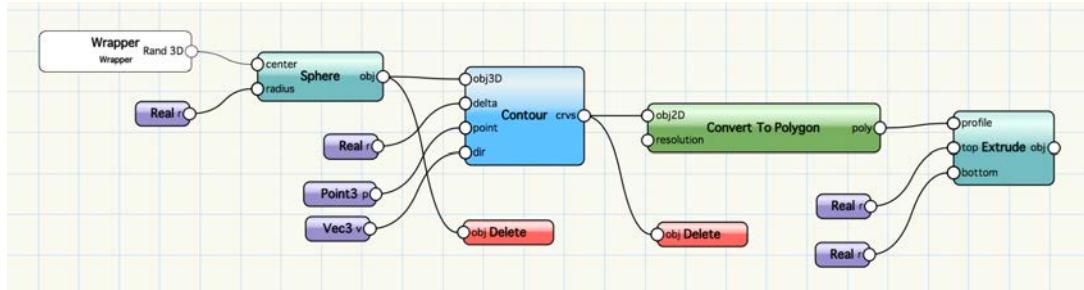
球状の構造物をひとつつくるマリオネットワークを編集して、複数の球の構造物を作成します。



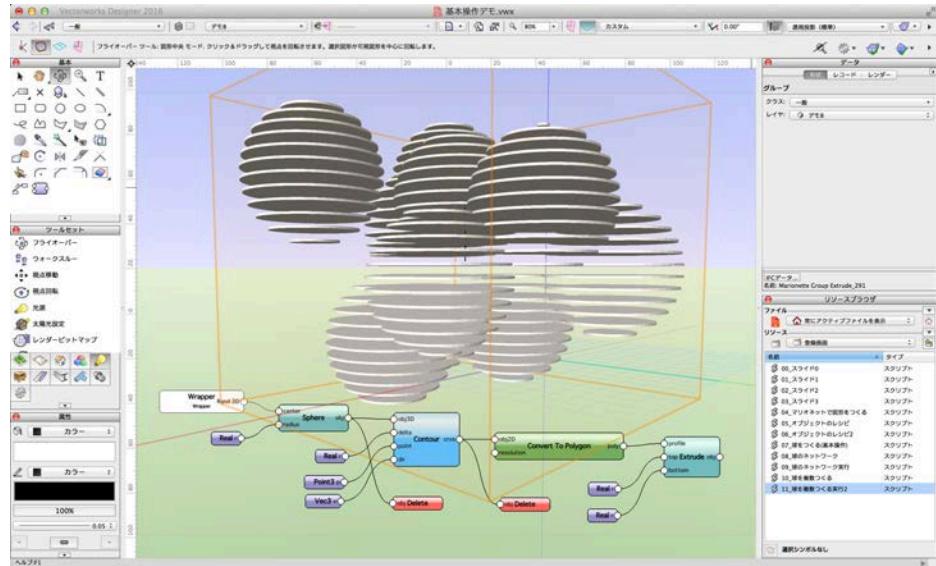
- 1) ランダムな座標を任意の数入力するノードを準備します。このノードはデフォルトで 10 個の座標を出力します。



- 2) ひとつの球の構造物をつくるネットワークに、ランダムな座標をつくるノードを接続します。ここでは、「Sphere」ノードの球の中心座標を設定する"center"に接続します。

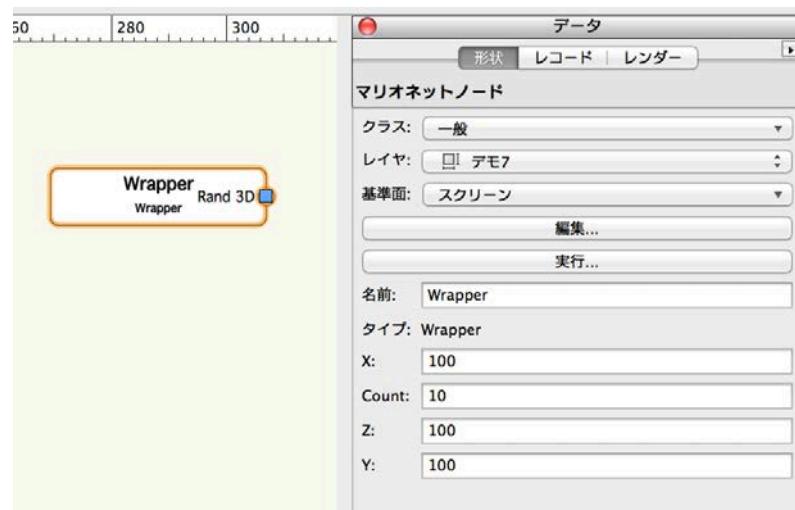


- 3) 実行ボタンを選択すると、「Sphere」に入力された座標の数だけネットワークが繰り返し実行され複数の球の構造物が作図されます。

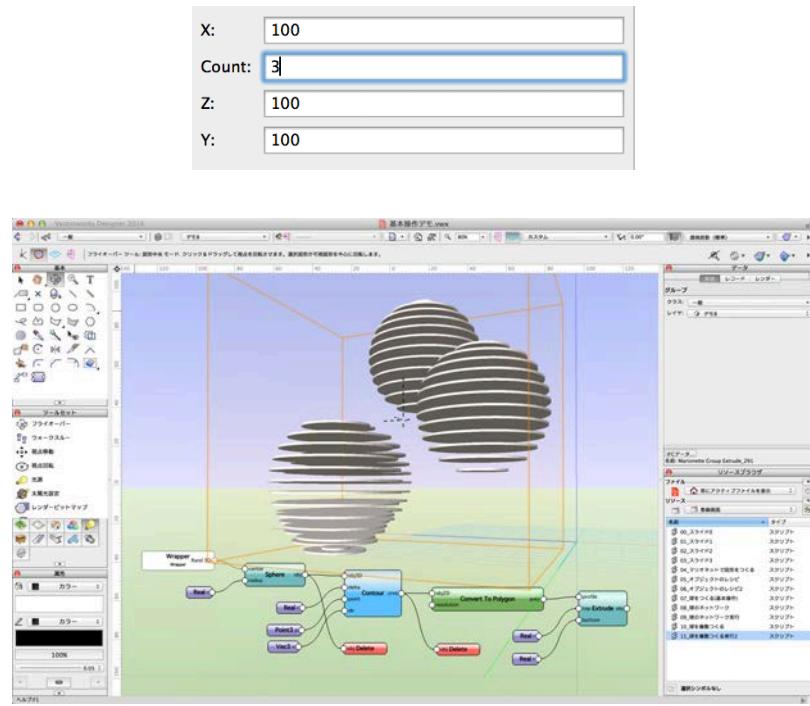


#### 1-4. ノードのパラメータを変更する

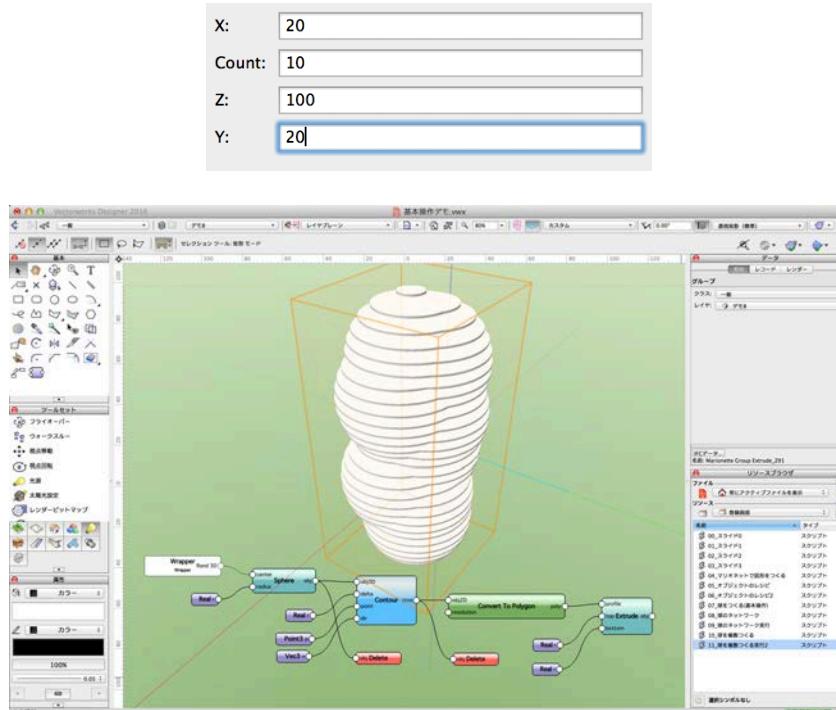
- 1) 今回使用している「ランダム座標を生成する」ノードは、データパレットから「座標の数」と「座標の範囲」を設定できます。ひとつのマリオネットワークからパラメータの設定によって異なる図形を作図することができます。



2) 生成する座標の数を変更して再度実行します。ここでは"Count"のパラメータを"3"に変更します。



3) 座標の範囲を変更して再度実行します。ここでは"X"と"Y"のパラメータをともに"20"に設定します。

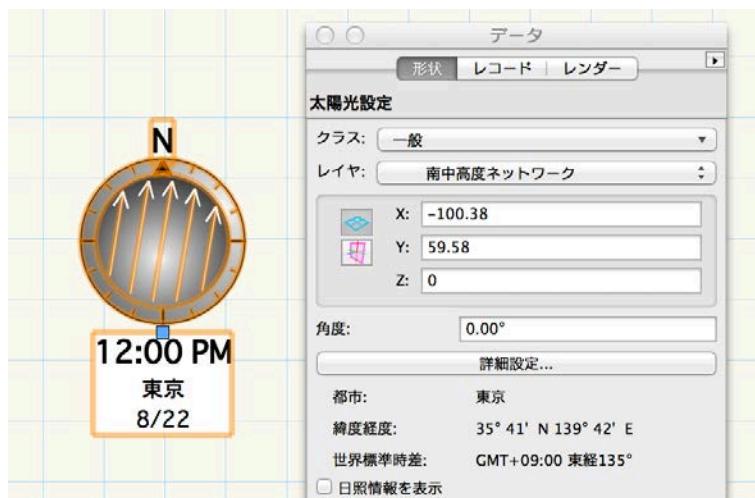


## 2. 南中高度を計算する

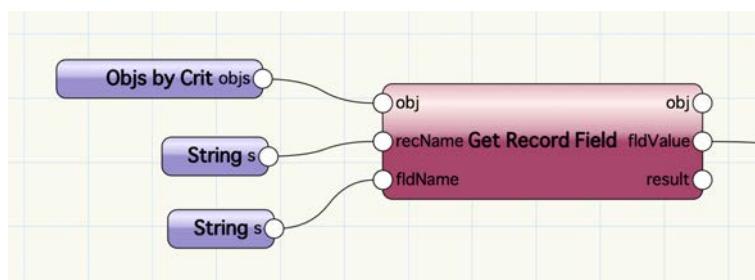
### 2.1. 太陽光設定オブジェクトから緯度の情報を取得する

マリオネットでは、図面上のプラグインオブジェクトからパラメータの値を取得してプログラムに利用することができます。ここでは太陽光設定オブジェクトから緯度の情報をマリオネットで扱えるデータとして取得します。

- 1) 図面上に太陽光設定オブジェクトを配置します。



- 2) マリオネットでは「Get Record Field」ノードを用いて、図面上の他の図形(プラグインオブジェクト)に設定してあるパラメータを取得することができます。"obj"に図形のハンドル、"recName"にプラグインオブジェクト名、"fldName"にパラメータの名前を入力します。



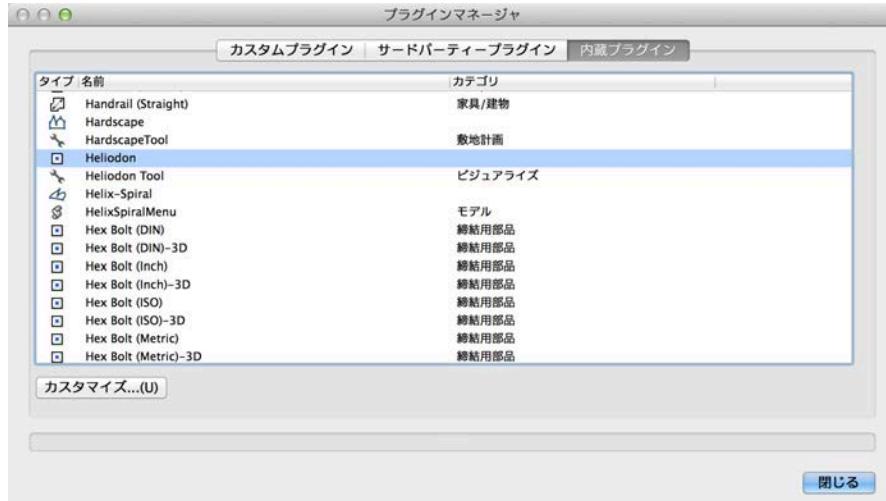
- 3) 太陽設定オブジェクトの緯度パラメータを取得するため、次のように入力します。

- ・ プラグインオブジェクト名 : Heliodon
- ・ パラメータ名 : \_\_Latitude

【参考】パラメータ名称を調べる

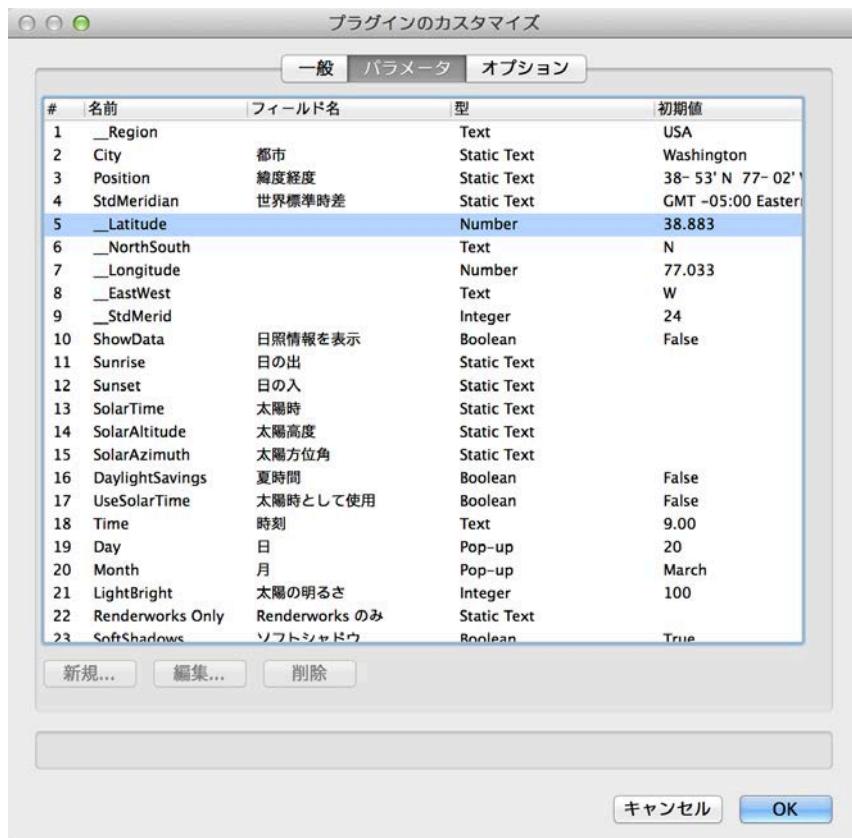
[1] プラグインマネージャをひらきます（ツール > プラグイン > プラグインマネージャ）

[2] 一覧から目的のプラグインオブジェクトを選択します



[3] プラグインのカスタマイズダイアログをひらきます

[4] 目的のパラメータの英語名を調べます



- 4) 図面上の太陽光設定オブジェクトのハンドルを取得するために「Obj by Crit」ノードを使用します。このノードはデータパレットの“Criteria”フィールドに図形の検索条件を入力すると、検索条件にあったすべての図形のハンドルを取得します。
- ここでは「種類=太陽光設定オブジェクト(Heliodon)」と「同じレイヤ(南中高度ネットワーク)にあるオブジェクト」を条件とするため「((L='南中高度ネットワーク') & (PON='Heliodon'))」と入力します。



#### 【参考】検索条件式をつくる

図形選択マクロをつかって、「Obj by Crit」ノードの条件式を作成することができます。

- [1] 図形選択マクロで、“Python スクリプトを作成”を選択します。



- [2] 検索条件を設定します。

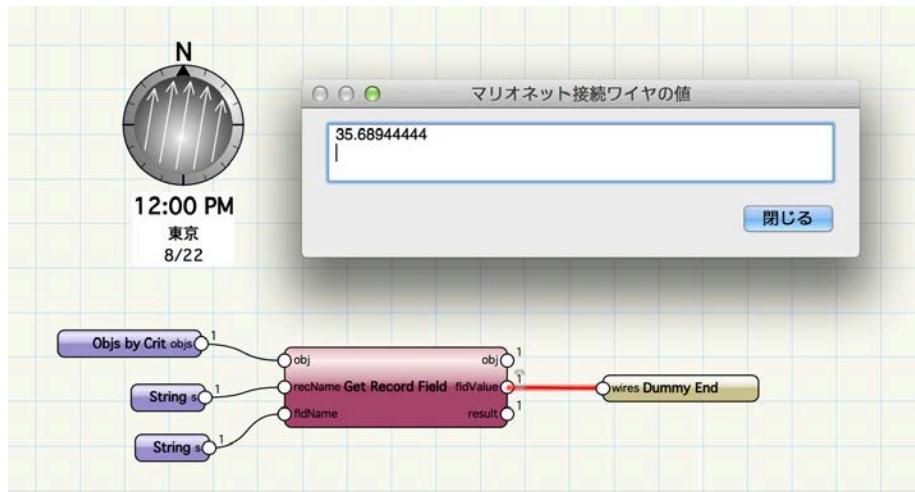


[3] 作成された Python スクリプトの編集画面ひらいて、条件式をコピーします。



```
import vs;
vs.SelectObj("((L='南中高度ネットワーク') & (PON='Heliodon'))");
```

5) 取得した緯度をデバッグモードで確認します。



## 2.2. 冬至の南中高度の計算式をつくる

数学関数カテゴリのノードを使用して、冬至の太陽の南中高度の計算式をつくります。

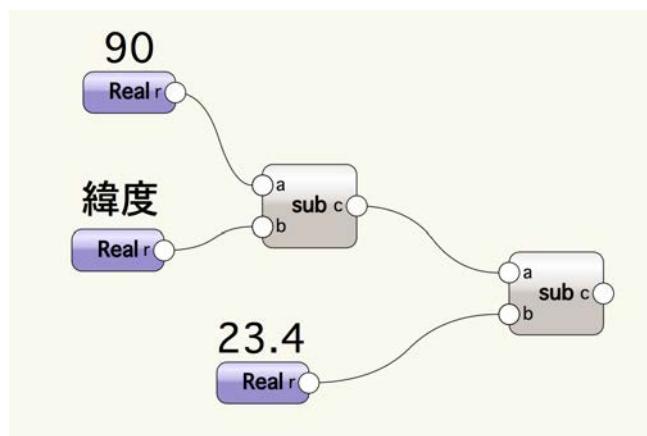
- 1) 冬至の南中高度を求める計算式をマリオネットで作成します。

$$\text{冬至の太陽の南中高度} = 90 - \text{緯度} - 23.4$$

マリオネットで計算式を表現するには一覧から数学関数カテゴリのノードを選択します。引き算を行う「Sub」ノードを選択します。

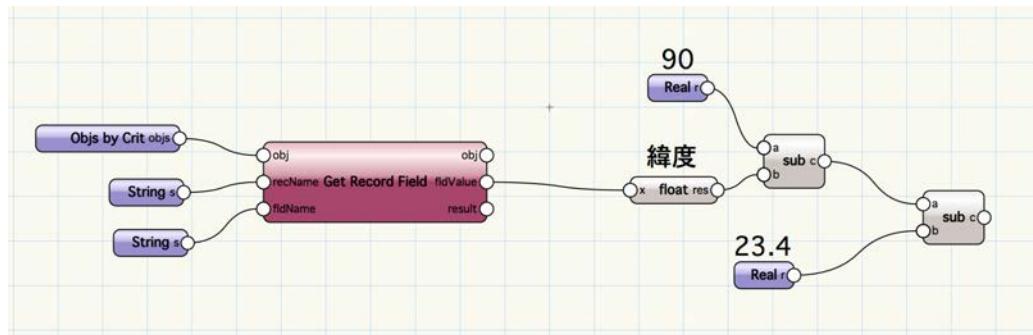


- 2) 「Sub」ノードは「"a"- "b"」の結果を返します。ひとつめの「Sub」で「90 - 緯度」ふたつめの「Sub」で「ひとつめの結果 - 23.4」を表現します。

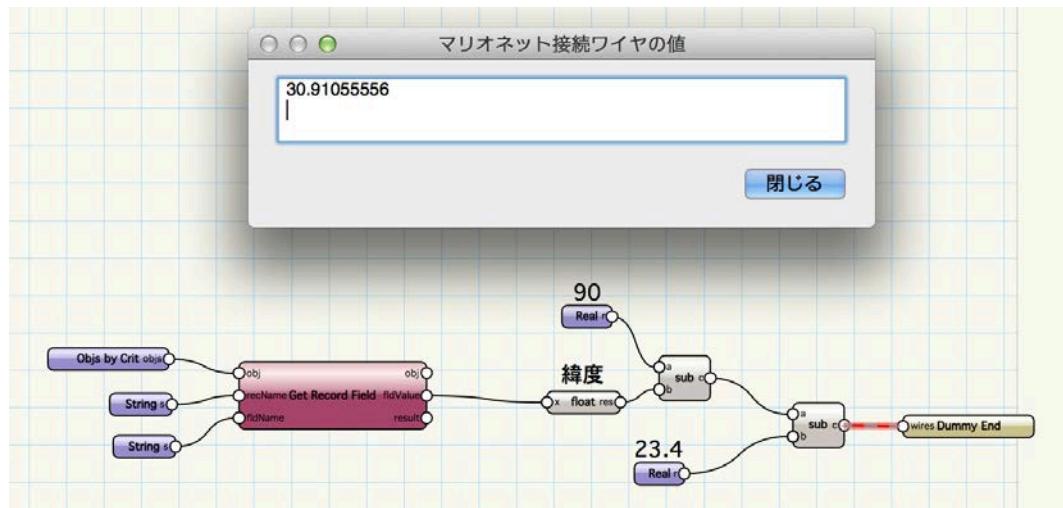


### 2.3. 緯度の値を南中高度の計算式で利用する

- 1) 2.1.で作成した緯度のパラメータを取得するネットワークと、2.2.で作成した南中高度の計算式を連動させて、緯度の値を南中高度の計算に利用します。「Get Record Field」ノードで取得するパラメータはすべて文字列になっているため、計算式で扱えるように「float」ノードを用いて実数値に変換したあと、南中高度の計算式に入力します。



- 2) 南中高度の計算結果をデバッグモードで確認します。



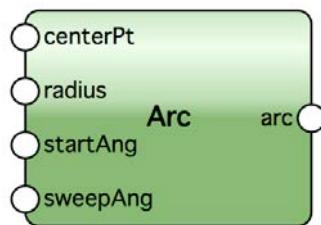
### 3. ワークシートから風配図をつくる

風向の 16 方位分の円弧図形を作成し、各円弧図形の半径を風向の頻度と対応させることで風配図を表現します。風向のデータはあらかじめワークシートに用意したものを使用します。

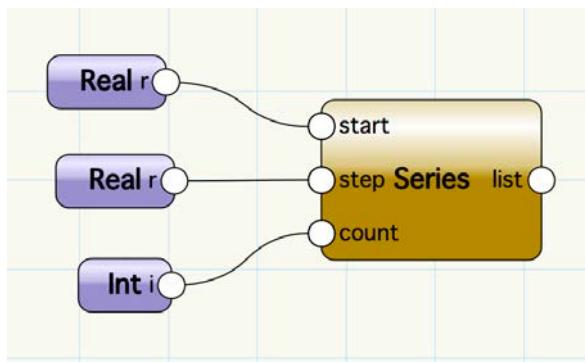
#### 3.1. 繰り返し処理を用いて円弧図形を 16 個作成する

風配図のベースとなる 16 個の円弧を作成します。

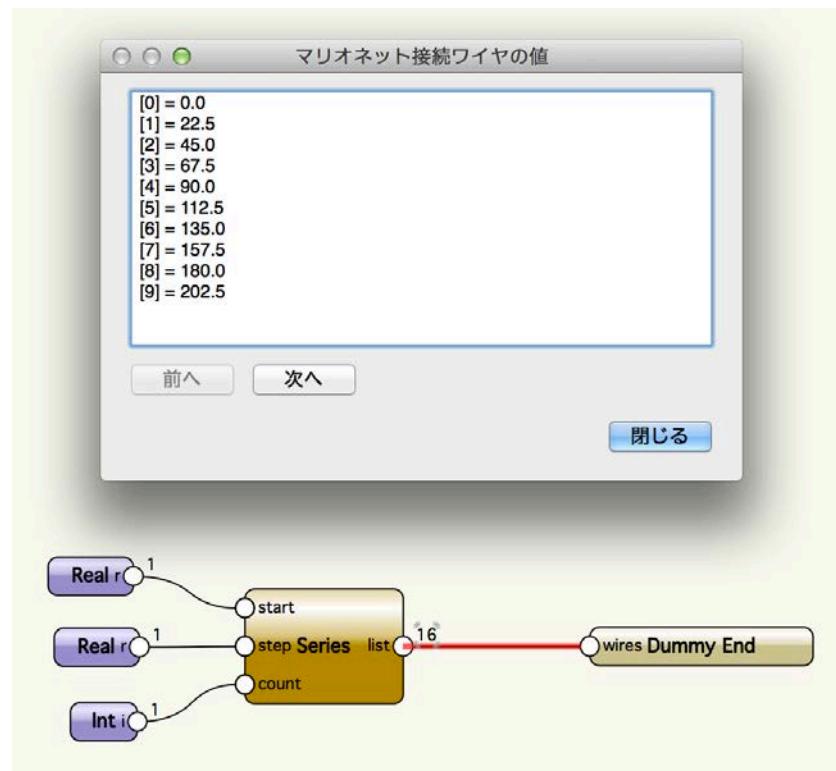
- 1) 16 方角分の円弧を描くには、円弧図形を作図する「Arc」ノードを 16 回繰り返します。  
「Arc」ノードの"startAngle"ポート・"sweepAng"ポートに 16 方位の角度のリストを入力することで、「Arc」が角度を変えながら 16 回繰り返し実行されます。



- 2) 「Arc」ノードに入力する 16 方位の角度のリストを「Series」ノードを用いて用意します。  
「Series」ノードは"start"の値に"step"の値を"count"回だけ加算しながら値のリストを作成します。"start"に Real ノードで 0 度、"step"に Real ノードで 22.5 度(360 度の 16 分割)、"count"に Int ノードで 16 を入力します。

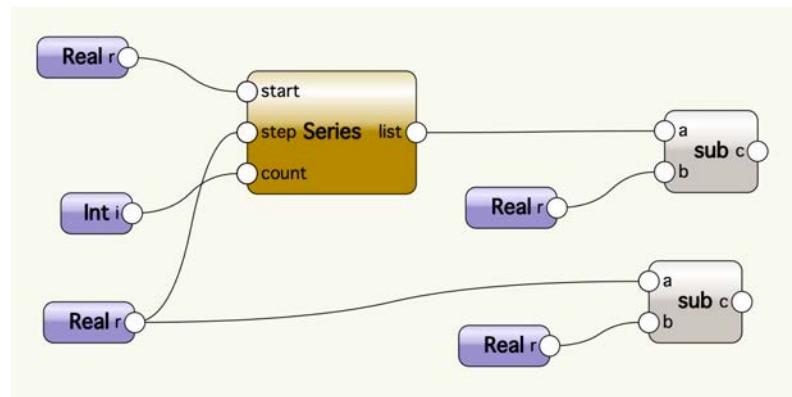


- 3) 作成した角度のリストをデバッグモードで確認します。「Series」ノードから 16 方位の角度のデータが出力されているのを確認できます。

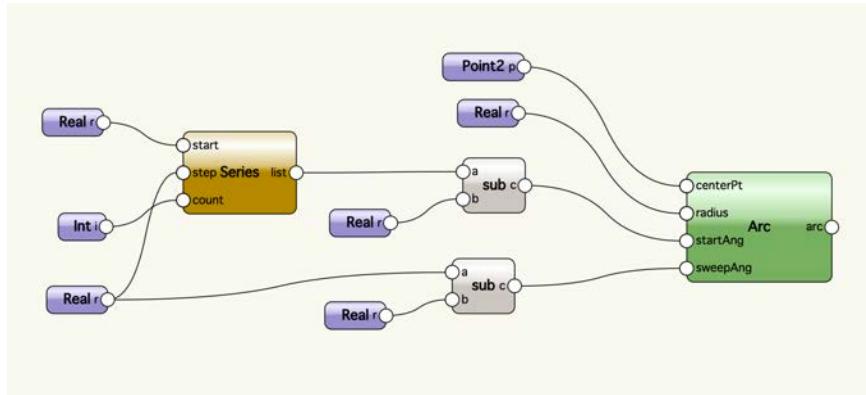


- 4) 「Arc」ノードに入力する円弧の開始角度(StartAng)と円弧角(SweepAng)の値のリストを作成します。ここでは円弧角を 18.5 度として各方位の角度から 16 個の円弧図形それぞれの開始角度を計算します。引き算のノード「Sub」を用いて計算します。

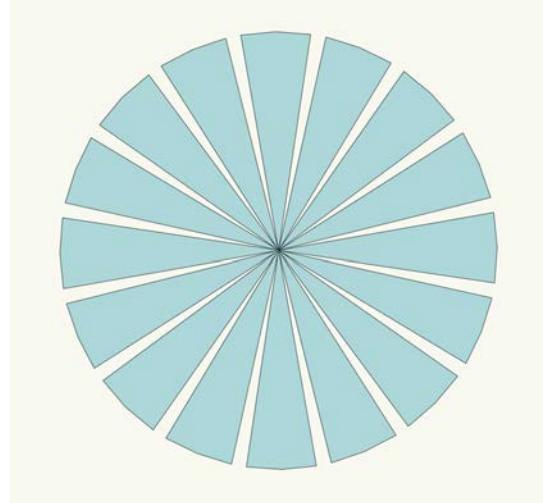
- 開始角度 = 方位角 - 8.5
- 円弧角 = 22.5 - 4



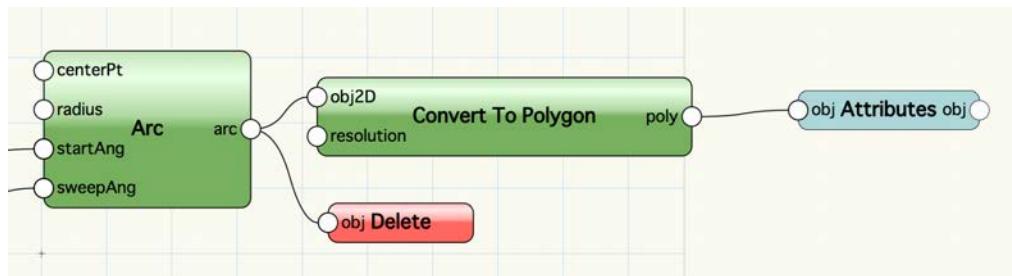
5) 作成した値のリストを「Arc」ノードの"startAngle"と"sweepAng"に接続します。



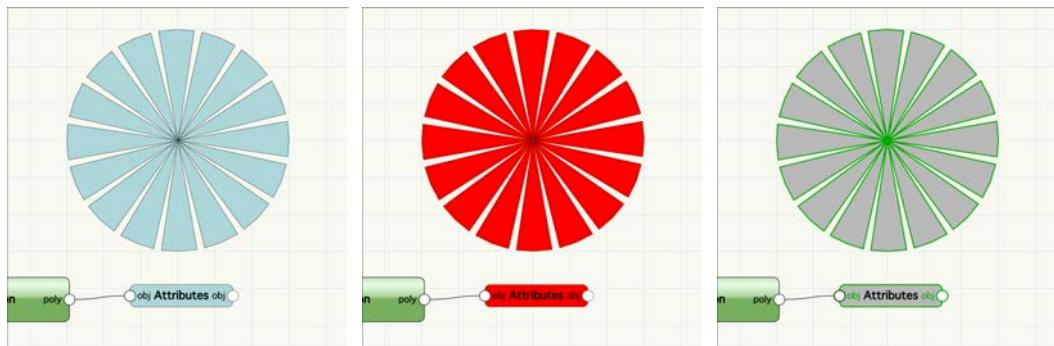
6) 「Arc」ノードが角度を変えながら 16 回の処理を行い、方位ごとに円弧が作成されます。



7) 作成した円弧の見た目を調整します。円弧図形に輪郭線を表示するために「Convert to Polygon」ノードを用いて多角形に変換します。「Convert to Polygon」ノードは変換した元の図形が図面上に残るため「Delete」ノードにつないで削除しておきます。



- 8) 変換した図形に「Attribute」ノードで色を付けます。「Attribute」ノードの図形に Vectorworks 上の図形と同じように面の色や透明度等の属性を設定しておくと、このノードを通過するマリオネットの図形の属性にそのまま反映されます。



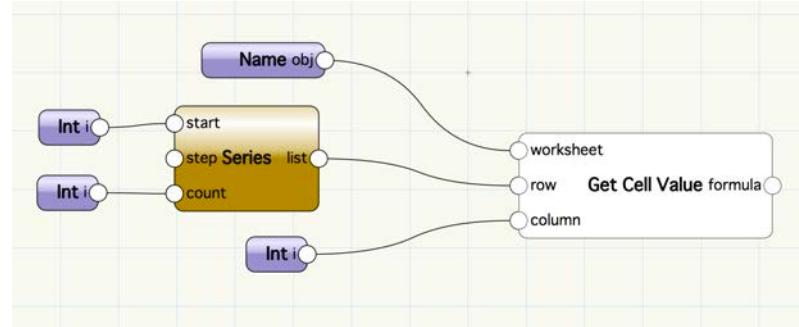
### 3.2. 円弧図形とワークシートの風向データを連動する

ワークシートにあらかじめ入力した 16 方位の風向データから各方位の頻度を抽出し風配図に反映させます。

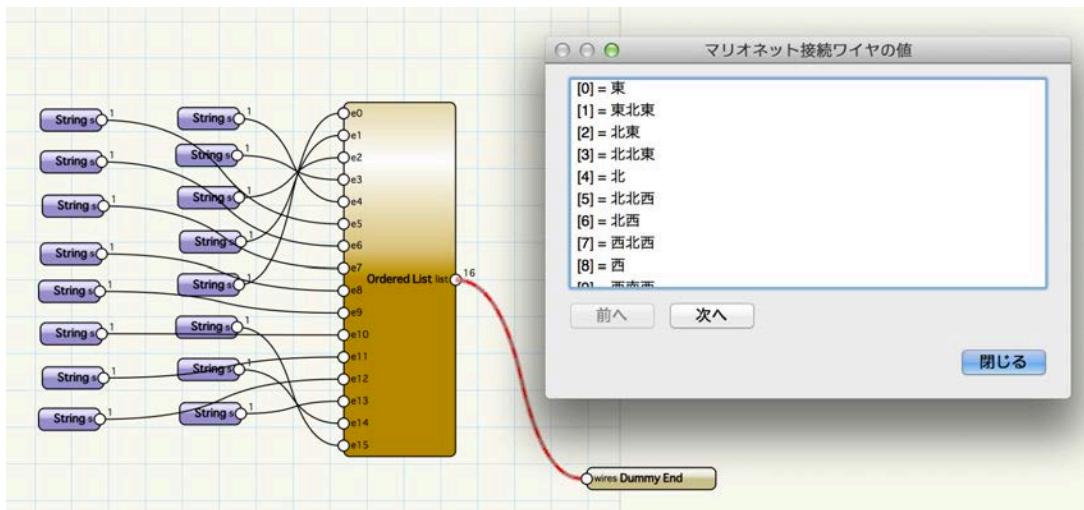
- 1) 風向のデータを記録したワークシートを用意します。ここでは東京 8 月 1 時間毎 5 年分の風向データを用意しています。

風向風速8月 @ 150%		
	A	B
1	▶ 東京	東京
2	▶ 風速 (m/s)	風速 (m/s)
3	▶	風向
4	▶	
5	▶	3.7 南南西
6	▶	3.5 南西
7	▶	3.3 南
8	▶	2.1 南
9	▶	2.4 南南西
10	▶	2.9 南南西
11	▶	3.1 南西
12	▶	2.2 西南西
13	▶	1.7 南南西
14	▶	2.9 南西
15	▶	5.2 南
16	▶	3.7 南南西
17	▶	4.5 南南西
18	▶	4.3 南
19	▶	4.7 南
20	▶	4.3 南
21	▶	3.9 南南西
22	▶	4.4 南南西
23	▶	4.8 南
24	▶	3.8 南
25	▶	2.4 南南西
26	▶	2.2 南南西
27	▶	3.2 南南西
28	▶	2.3 南南西
29	▶	4.2 南南西
30	▶	4.4 南南西
31	▶	2.2 南南西
32	▶	3.7 南
33	▶	3.8 南

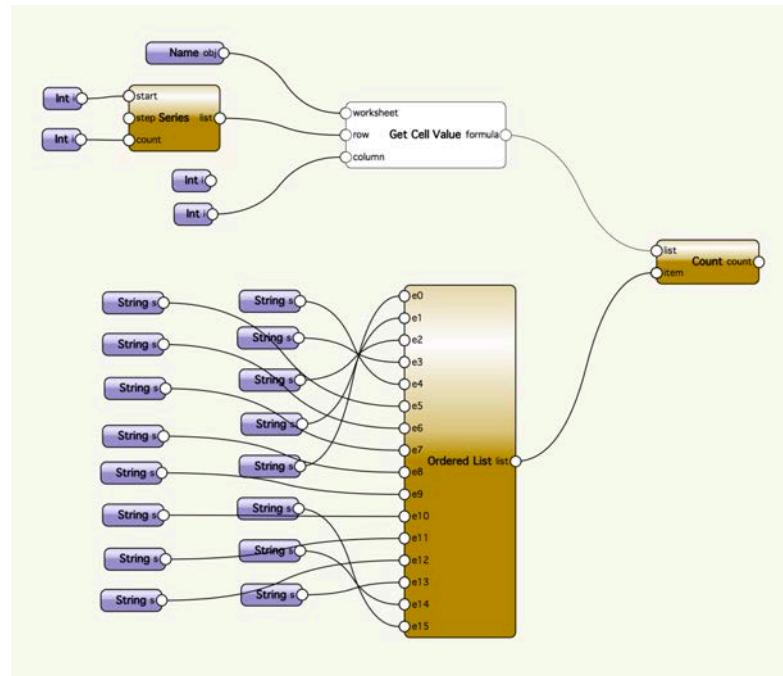
- 2) ワークシートから「Get Cell Value」ノードを用いて風向データを取得します。「Get Cell Value」ノードにワークシートの名前・列番号・行番号のリストを渡し、ワークシートの風向データをすべて取得してリスト化します。



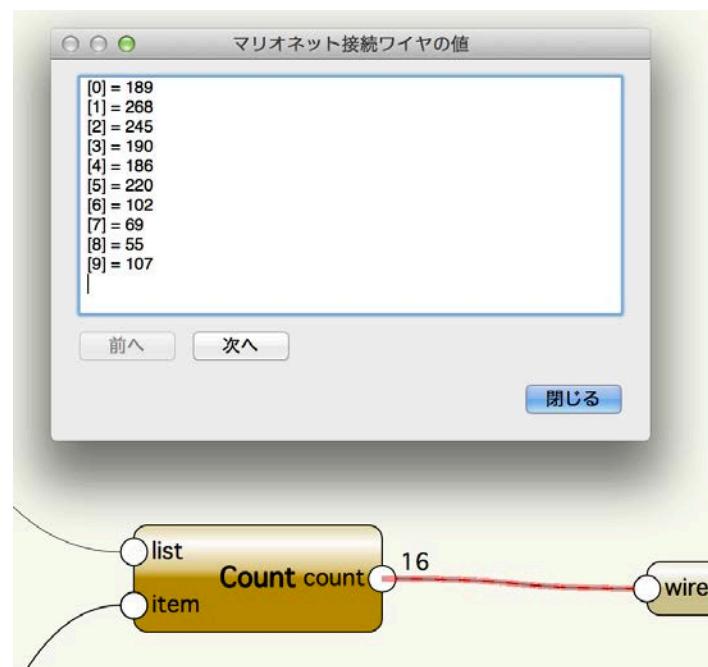
- 3) 作成した風向データのリストから各方位がいくつ含まれているか数えます。準備として 16 方位の名前のリストを作成します。この名前のリストは東を先頭に半時計回りで東南東までの順番で作成する必要があります。これは方位と 3.1.で作成した角度のリスト(図面上での 0~360 度)を合致させるためです。



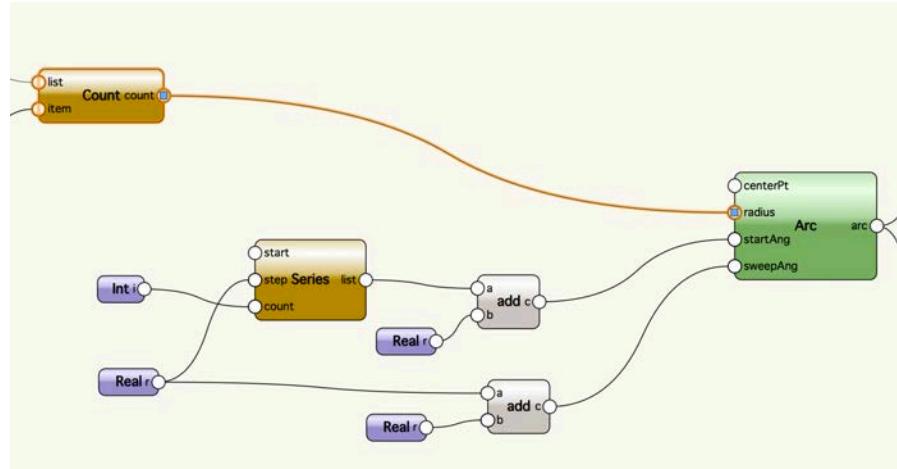
- 4) 風向データのリストと 16 方位の名前のリストを「Count」ノードに入力して風向データから各方位の頻度を抽出します。「Count」ノードは“list”に入力されたリストの中に、“item”に入力された項目がいくつあるかを数え上げます。



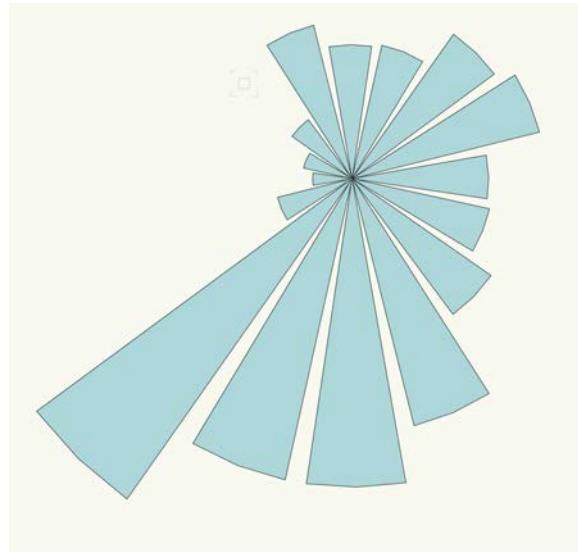
- 5) 抽出した頻度は「Count」ノードの“item”に入力された方位の名前のリストの順にリスト化されます。



- 6) 抽出した各方位の頻度の値を 3.2.で作成した円弧の半径に反映させます。「Count」ノードで算出した風向の頻度のリストを、16 方位の円弧を作成する「Arc」ノードの"radius"ポートへ入力します。



- 7) 「Count」で算出した風向の頻度が、円弧図形の半径に反映され風配が表現されます。

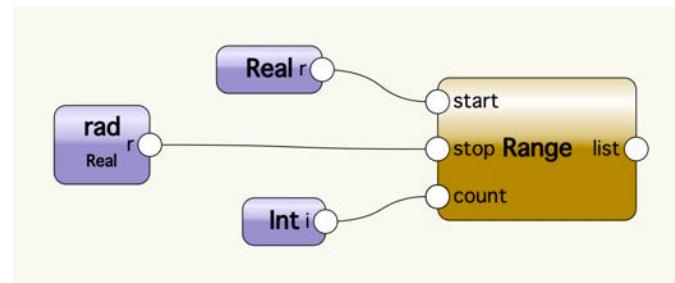


### 3.3. 目盛りや軸を作図して風配図の見た目を整える

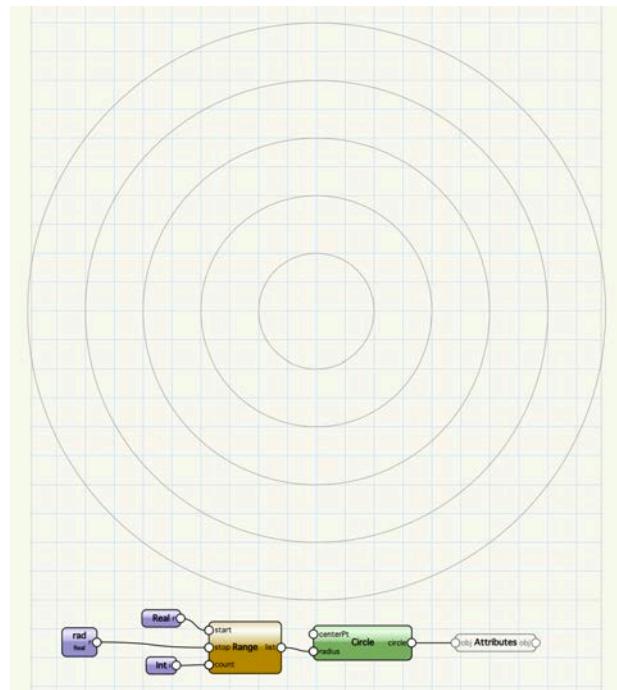
円図形や文字図形を使用して目盛りや軸の名前を作成します。

- 1) 風配図見た目を整えるため、ここでは文字図形で 4 方位(東西南北)の名前と目盛りの値、円図形で目盛り線、直線図形で 4 方位の軸を作成します。

- 2) 円图形で目盛りの線を作成します。風配図全体の半径をあらかじめ設定しておき、その半径を等分割する目盛り線を作成します。値を当分割するには「Range」ノードを用います。「rad」ノードで入力した半径を等分割した値のリストをつくります。



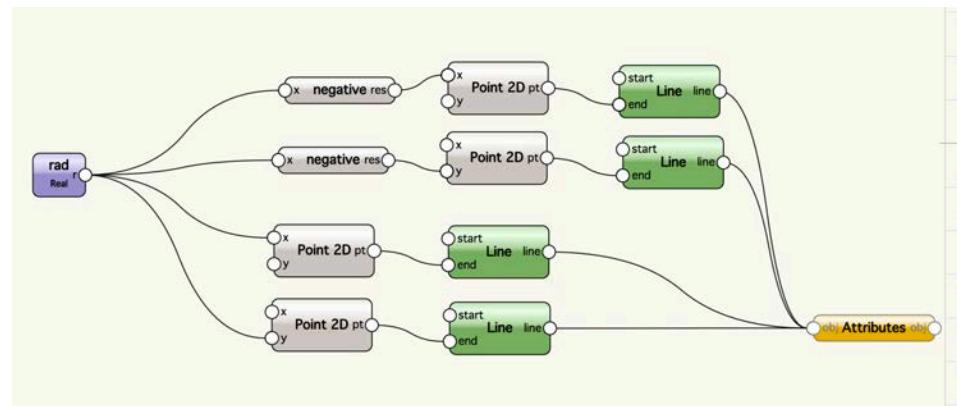
- 3) 「Range」で作成した値のリストを「Circle」ノードに入力することで、半径を分割間隔だけ増大しながら繰り返し円を作図します。また、「Attribute」ノードに入力して面の塗りを透明にしておきます。



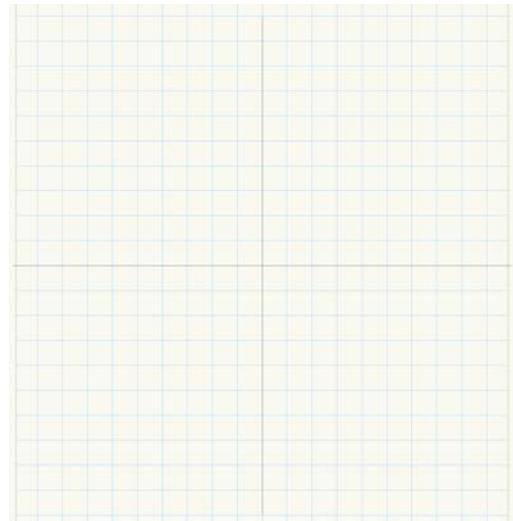
4) 直線図形で4方位の軸を作成します。4方位の軸の始点と終点の座標を半径の値をもとに設定します。

- ・ 東： 始点( 0, 0 ), 終点( 半径, 0 )
- ・ 西： 始点( 0, 0 ), 終点( -半径, 0 )
- ・ 南： 始点( 0, 0 ), 終点( 0, -半径 )
- ・ 北： 始点( 0, 0 ), 終点( 0, 半径 )

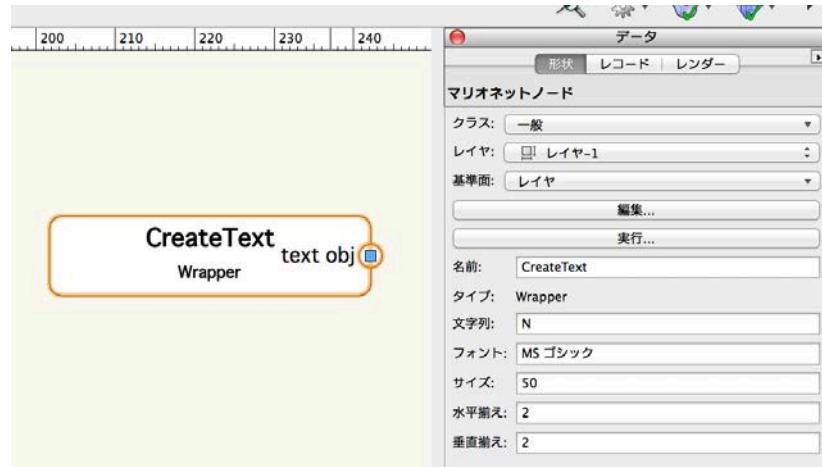
5) 4つの座標を「Point 2D」ノードに設定し、「Line」ノードに入力します。「negative」ノードを通過させることで符号を反転させて負の値を表現しています。



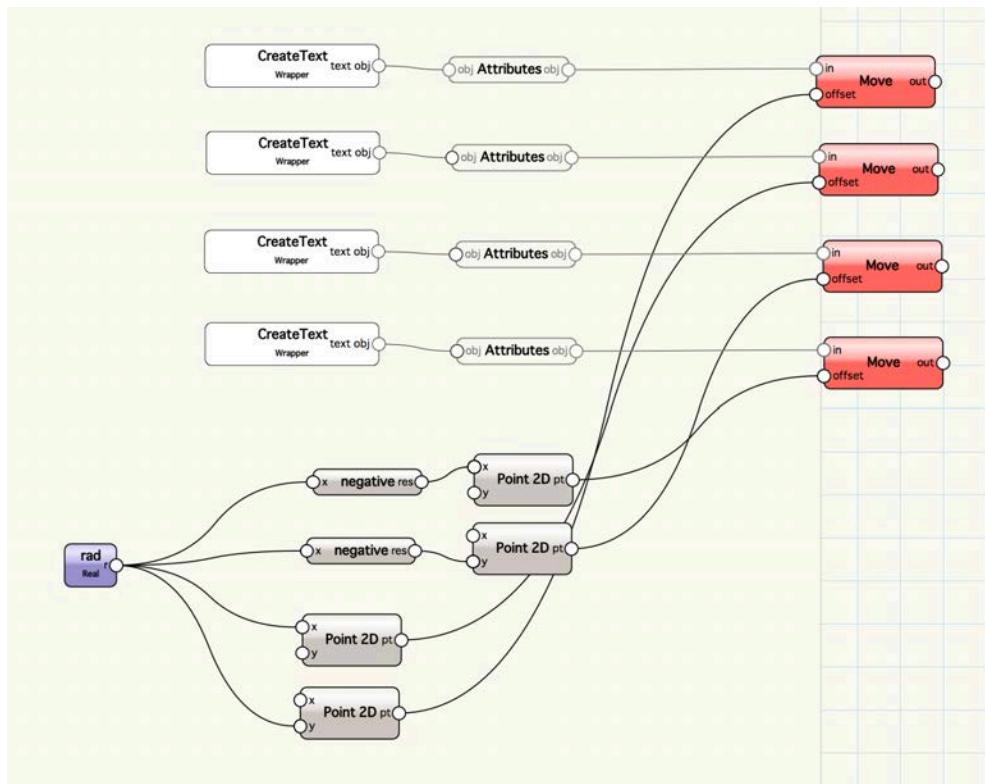
6) プログラムを実行すると4方向に軸が作図されます。



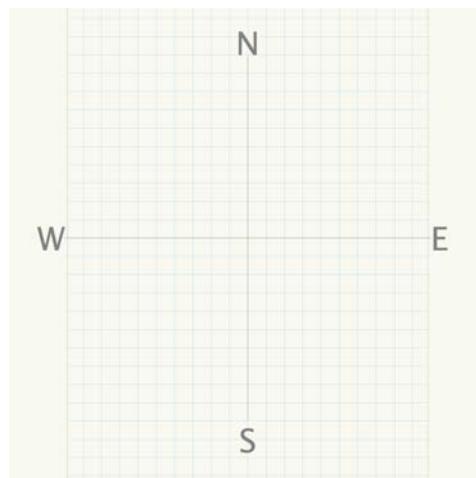
- 7) 文字図形で 4 方位の名前を作図します。ここではデータパレットに入力した文字を図面上に作成する「Create Text」ノードを用います。



- 8) 4 つの「Create Text」をつかって作成した 4 方位の文字図形を、軸の終点の座標に「Move」ノードで移動します。軸の終点の座標は 4) で作成した 4 方位のものを流用します。

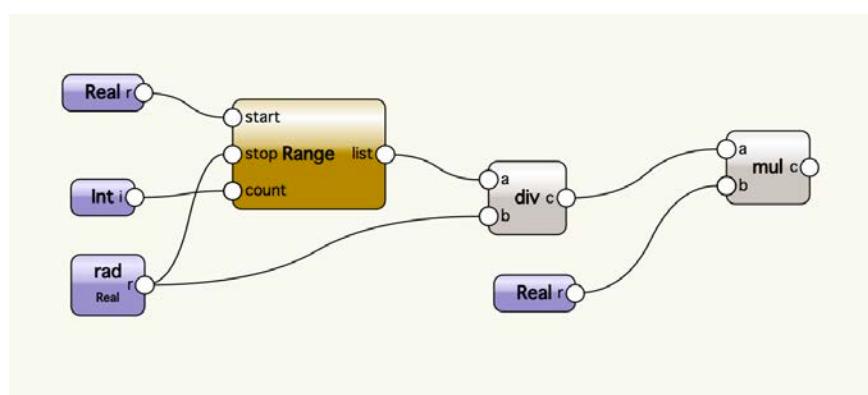


9) プログラムを実行すると 4 つの軸の終端に方位の名前が作図されます。

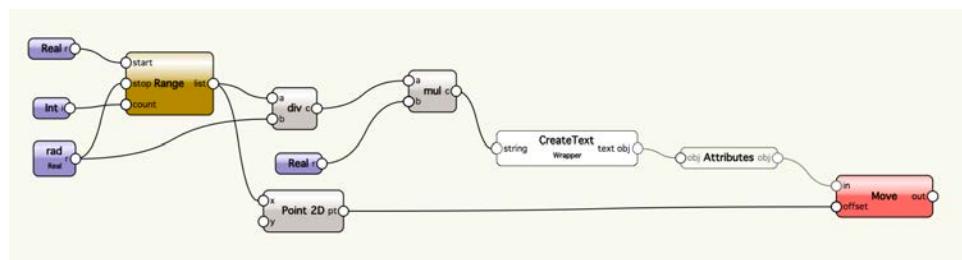


10) 文字図形を用いて目盛りの数値を表示します。一番外側の目盛り線が対応する風向頻度(最大頻度)を任意に設定し、最大頻度と目盛り線の半径から各目盛りの値を算出します。

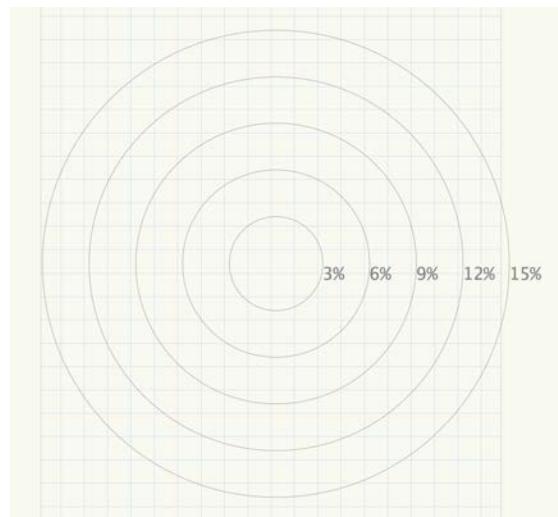
$$\text{目盛りの値} = \text{目盛り線の半径} / \text{最大半径} * \text{最大頻度(割合)}$$



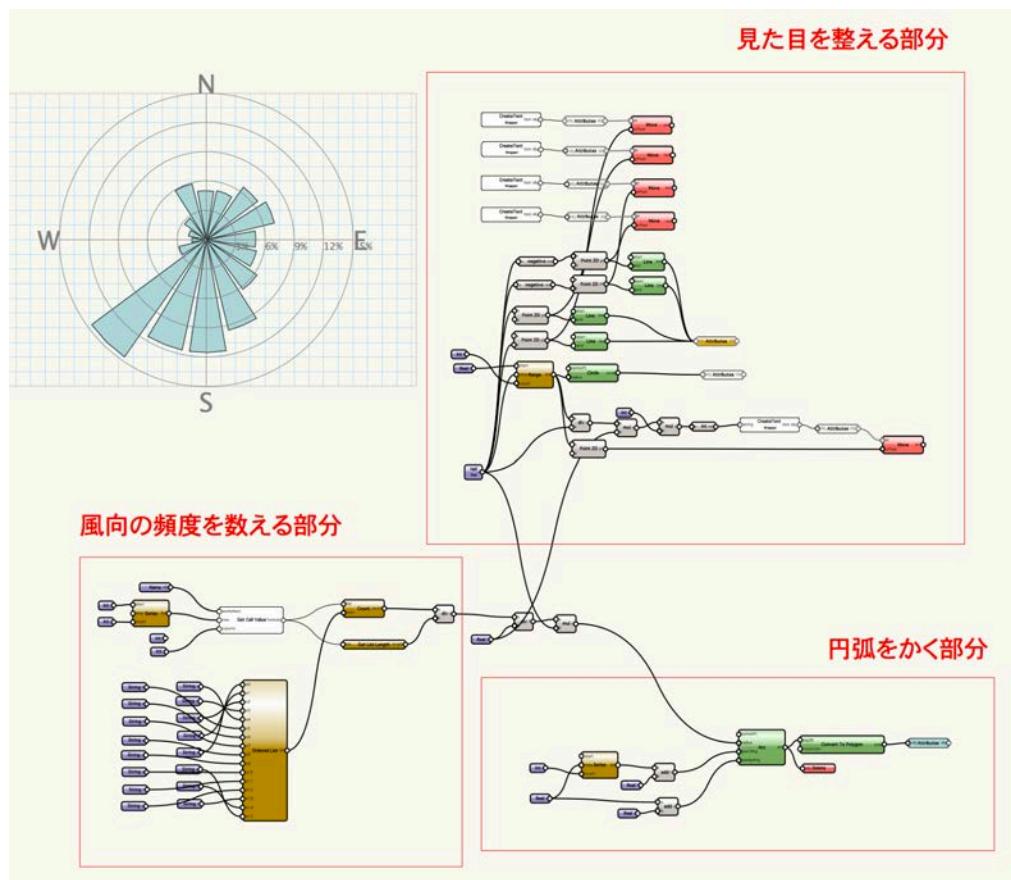
11) 目盛り値を文字図形で図面上に作図します。作図した目盛りの図形を各目盛り線の座標に「Move」ノードで移動します。目盛り線の座標値は「Range」ノードの結果を「Point 2D」ノードの"x"に入力して作成します。



12) プログラムを実行すると目盛り線の横に目盛りの値が作図されます。



13) 3.2.で作成した風向頻度を反映した円弧図形と、3.3.でここまでに作成した見た目を調整する図形を合体させて風配図の完成です。

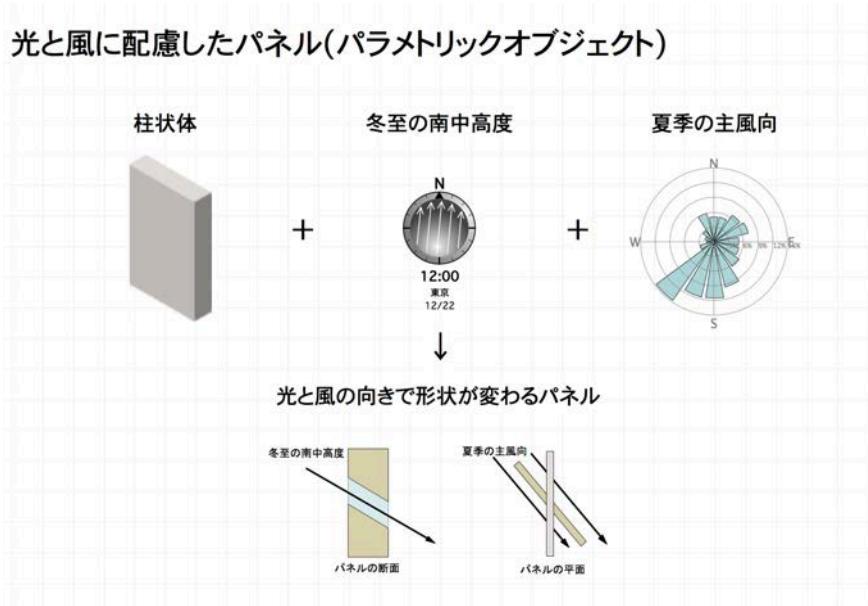


## 4. 光と風を考慮したパネルのパラメトリックオブジェクトをつくる

光の向きと風に向きを考慮したパラメータを持ち、そのパラメータによって形状が変化するパネルのオブジェクトを作成します。

柱状体をベースに 2 つのパラメータ(夏季の主風向、冬至の南中高度)に応じて、パネルの平面回転とスリットの角度を制御します。

光と風に配慮したパネル(パラメトリックオブジェクト)

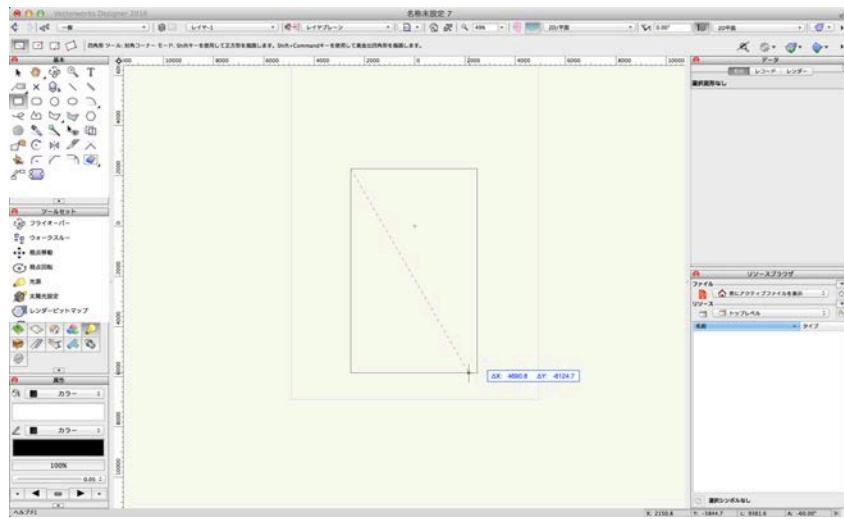


マリオネットオブジェクトを用いてパネルを作成します。マリオネットオブジェクトはネットワークを内包した特殊なオブジェクトで内部のネットワークの実行結果が見た目に反映されます。また、内包するネットワークの入力パラメータをオブジェクトのデータパレットに追加することができるほか、"コントロールジオメトリ"に図形の概形を保存することができます。

### 4.1. ファサードの概形をコントロールジオメトリに設定する

マリオネットオブジェクトでは、オブジェクトの概形(ベースとなる図形)をひとつ登録することができます。ここではパネルのファサードの概形をコントロールジオメトリに登録します。

- 1) 四角形または多角形でファサード概形を図面上に作図します。



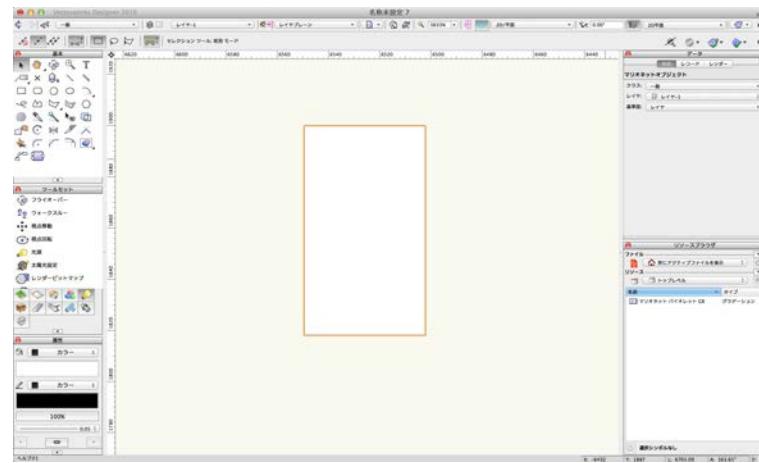
- 2) マリオネットツールで入力関数カテゴリから「Control Geometry」ノードを図面上に配置し  
ラッパーノードに変換します。



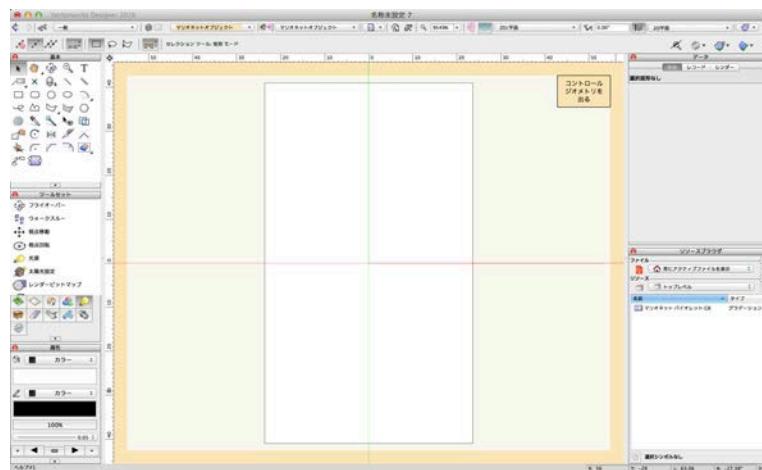
- 3) ラッパーノードと概形の図形を同時に選択し、右クリックでコンテキストメニューを開き「オ  
ブジェクトノードに変換」を実行します。



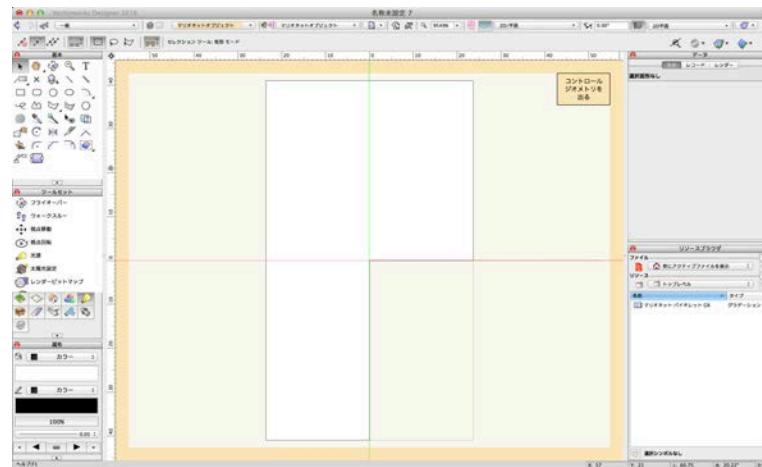
4) 四角形とラッパーノードがマリオネットオブジェクトに変換されます。



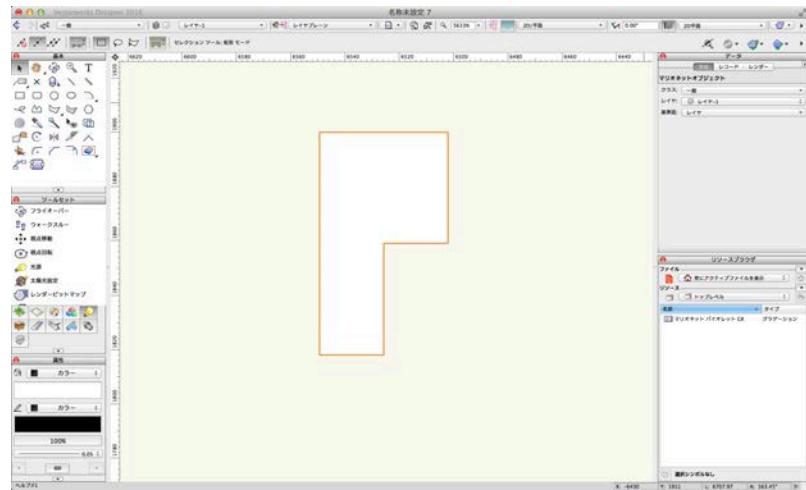
5) マリオネットオブジェクトを右クリック「コントールジオメトリを編集」を実行し、ファサー  
ドの概形が登録されていることを確認します。



6) コントールジオメトリの編集画面で図形を任意の形状に編集します。



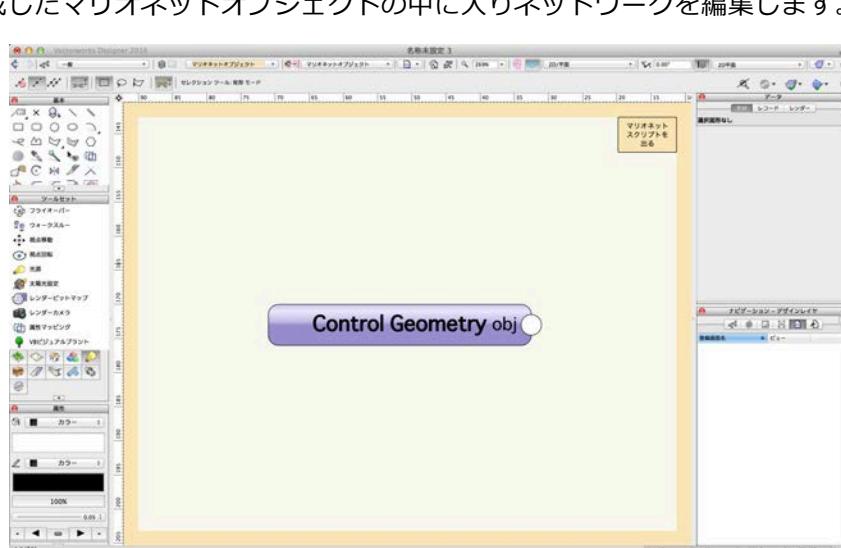
7) コントロールジオメトリの編集内容がマリオネットオブジェクトの見た目に反映されます。



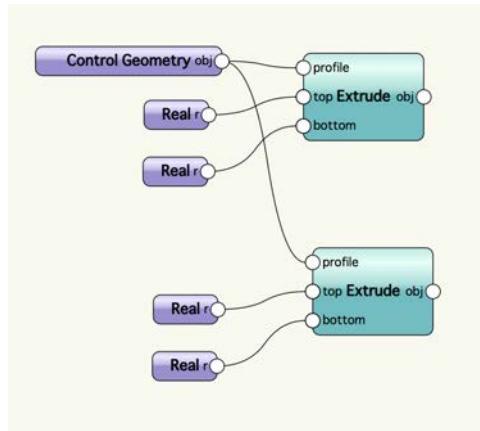
#### 4.2. パラメータに応じて柱状体を回転する

主風向や太陽角度のパラメータに応じて柱状体を回転させてパネルの形状をつくります。

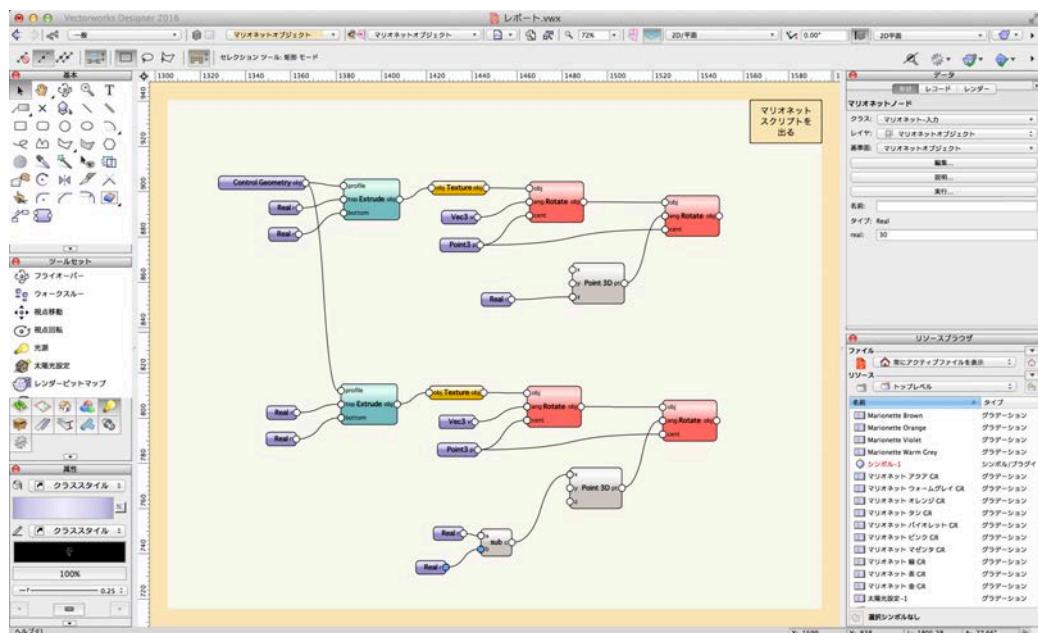
- 1) パネルの設計の条件「主風向パラメータの角度で平面回転する」と「南中高度パラメータでスリットを設ける」を満たすため、次のように考えます。
  - [1] 柱状体を 2 つ用意する
  - [2] 1 つの柱状体を主風向で回転し、もう 1 つを南中高度で回転する
  - [3] 主風向の柱状体を南中高度の柱状体で削り取りスリットを設ける
- 2) 4.1.で作成したマリオネットオブジェクトの中に入りネットワークを編集します。



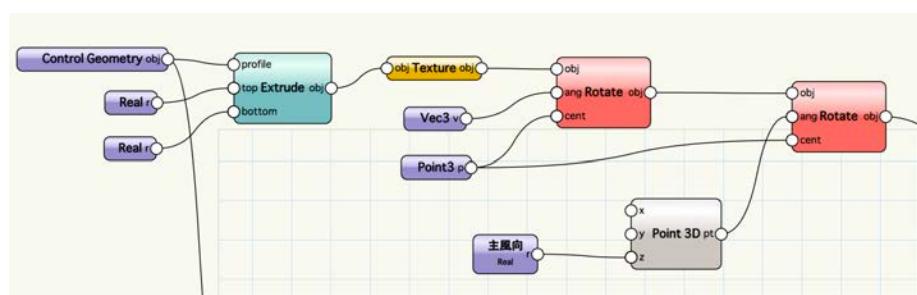
- 3) 「Control Geometry」ノードの出力からファサードが概形を取得し、その図形に厚みを与えてベースの柱状体とします。これを主風向のパラメータによって操作するためのものと、南中高度のパラメータで操作するため、2つ用意します。



- 4) 作成した柱状体を「Rotate」ノードを用いて主風向および、南中高度のパラメータで回転します。主風向はZ軸について、南中高度はX軸について回転します。



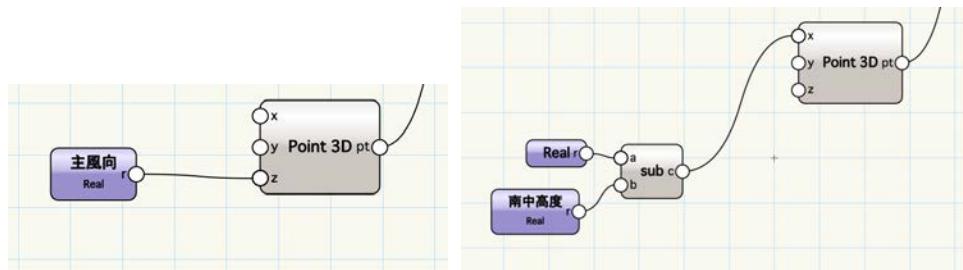
【補足】 「Point 3D」ノードに入力するパラメータを主風向は“z”ポート、南中高度は“x”ポートのように分けることで回転軸を指定することができます。



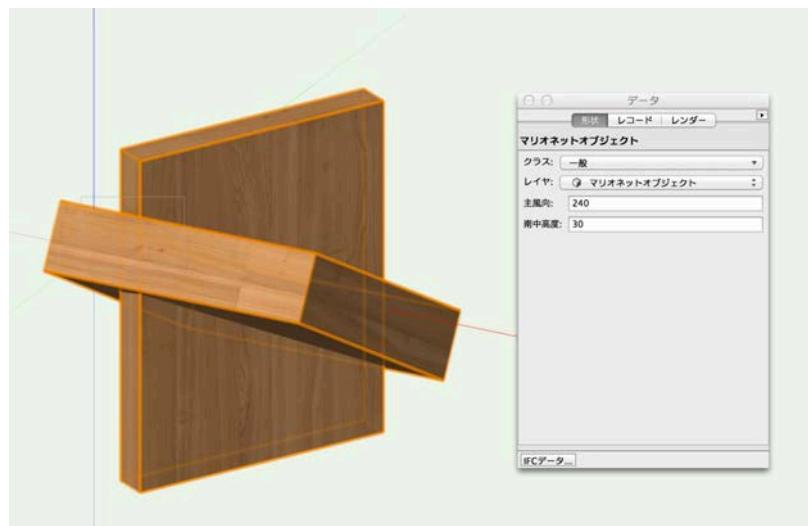
#### 4.3. データパレットにパラメータを表示する

主風向および南中高度のパラメータをマリオネットオブジェクトのデータパレットに表示します。

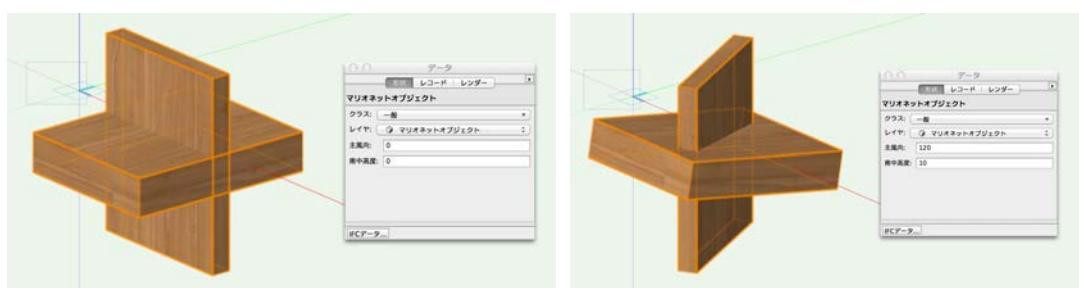
- 1) データパレットに表示するには内包するネットワークでパラメータ入力用のノードに名称を設定します。



- 2) 内部のネットワークで名前をつけたノードはマリオネットオブジェクトのデータパレットに表示されます。



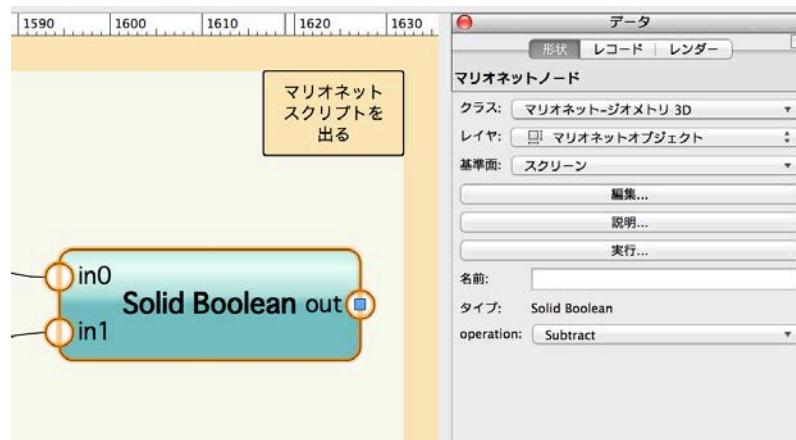
- 3) データパレットでパラメータを編集すると、内包するネットワークのプログラムが再度実行されてマリオネットオブジェクトの見た目に反映されます。



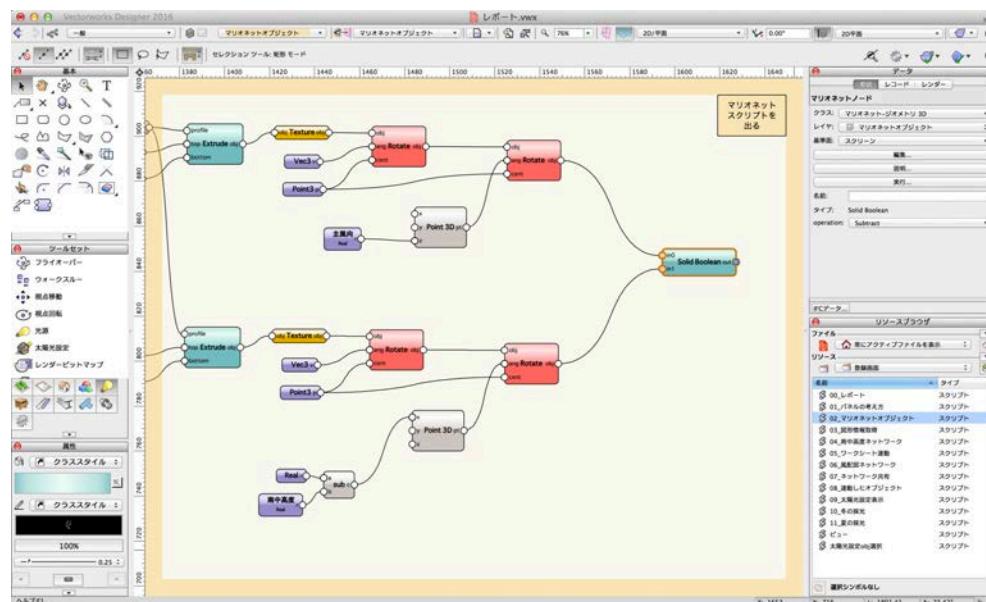
#### 4.4. パネルにガラスのスリットを設ける

パネルにスリットを設け形状を仕上げます。パネルのスリット部分をガラス部材の図形で作成します。

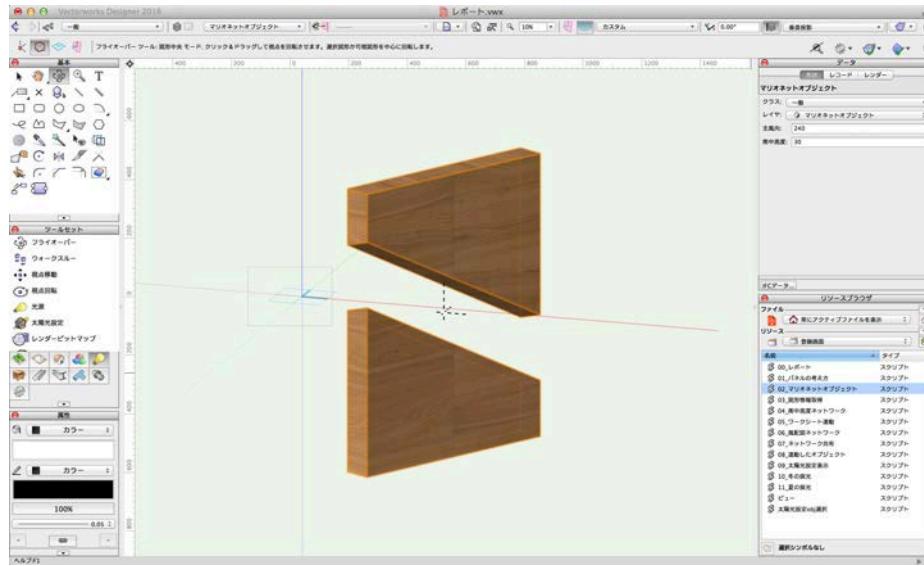
- 1) 2つの柱状体を削りとて、スリットが入ったパネルの形状をつくります。柱状体同士の削りとりは「Solid Boolean」ノードで行います。削り取りの場合「Solid Boolean」のデータパレットからオプションを"Subtract"に設定しておきます。



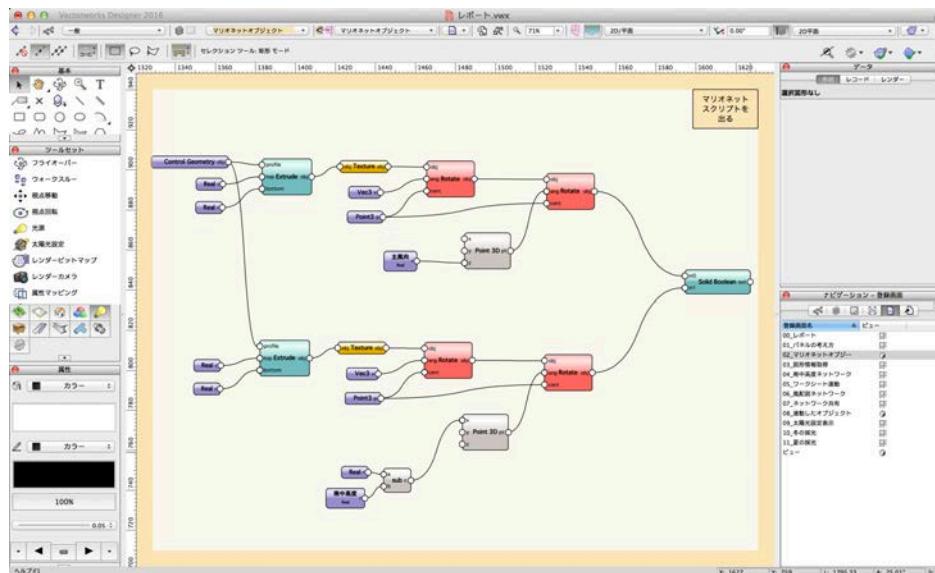
- 2) 回転した 2 つの柱状体を「Solid Boolean」ノードに入力します。“in0”に入力した風向の柱状体から、“in1”に入力した南中高度の柱状体を削り取ります。



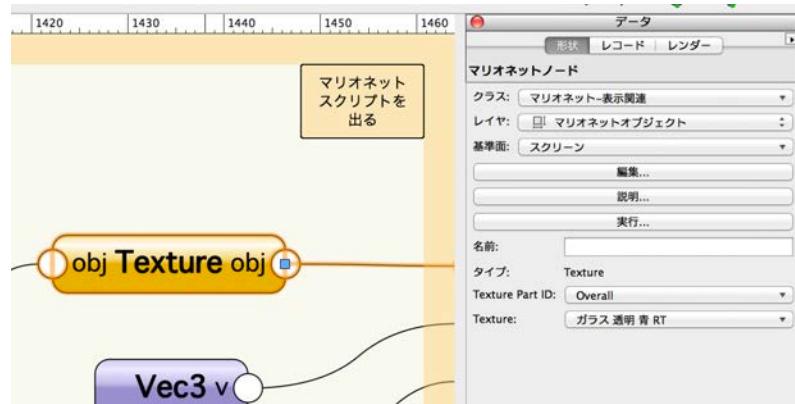
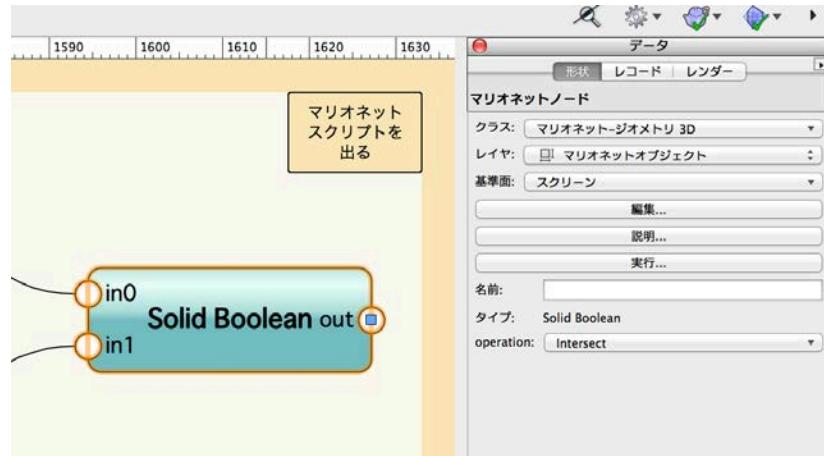
- 3) 削りとりの結果がマリオネットオブジェクトの見た目に反映されて、スリットの入ったパネルに変化したのを確認します。



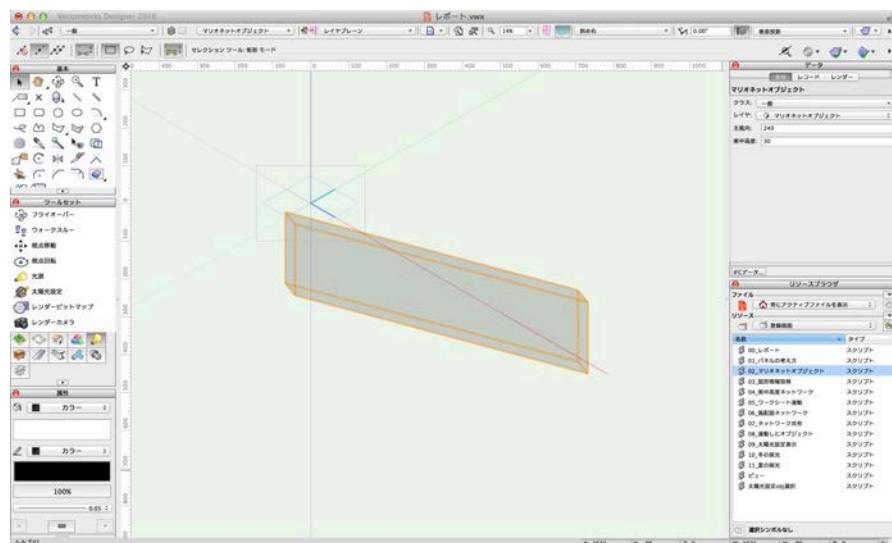
- 4) パネルのスリットの部分にはめこむガラスの部材を作成します。ガラスの部分の図形はここまで説明したパネルのネットワークと全く同じものを流用します。



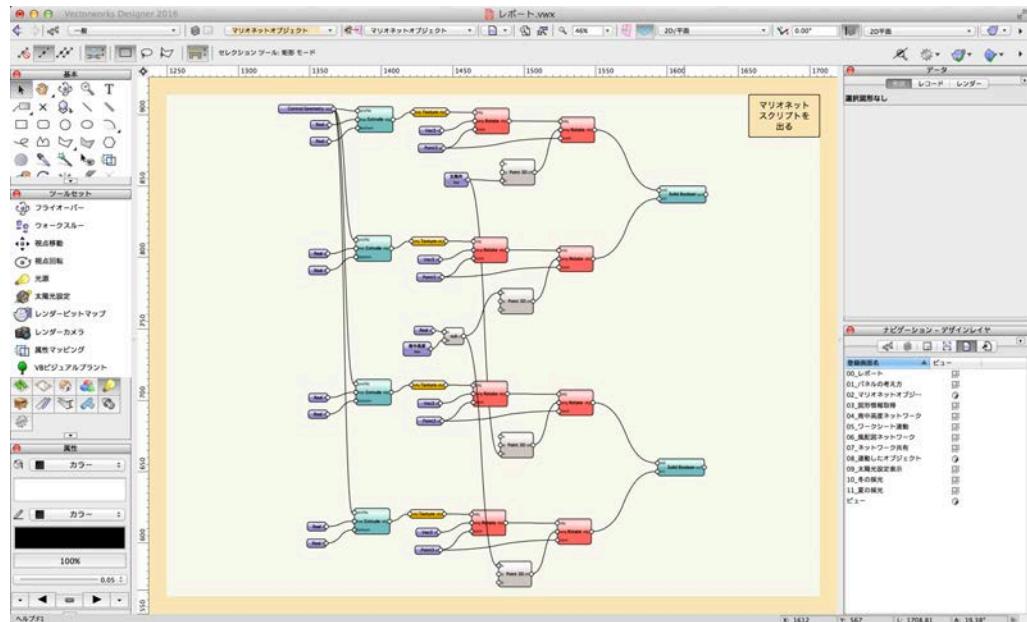
- 5) 使用するノードやワイヤーの接続順は全く同じですが、「Solid Boolean」ノードのオプションを"Intersect"に、「Texture」ノードのテクスチャを"ガラス"に変更します。



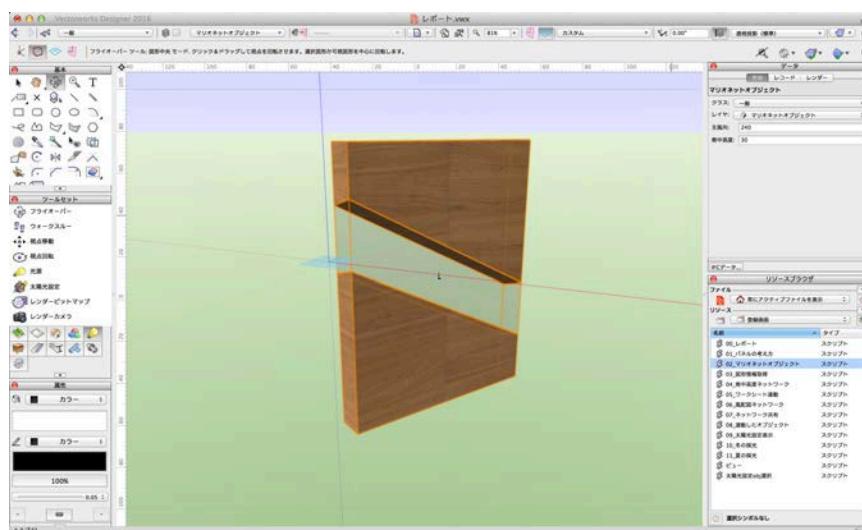
- 6) オプションを変更してプログラムを実行して(マリオネットスクリプトを出る)、ガラス部分のオブジェクトが作成されているのが確認します。



7) パネルのネットワークにスリットのガラス部分のネットワークを連結します。ここでは 2 つのネットワークで「Control Geometry」ノード、主風向および南中高度パラメータを入力する「Real」ノードを共有するように連結します。



8) プログラムを実行するとスリット部分にガラスが入ったパネルが作成されます。



## マリオネット関連情報

1. Vectorworks 社ユーザーフォーラム

<https://forum.vectorworks.net>

2. Marionette 全般

<https://forum.vectorworks.net/index.php?/forum/48-marionette/>

3. Marionette リソース集

<https://forum.vectorworks.net/index.php?/forum/47-resource-share-marionette/>

4. Marionette Tutorials

<http://kbase.vectorworks.net/questions/1350/Marionette+Tutorials>

5. Developer Wiki Marionette 詳細

<http://developer.vectorworks.net/index.php/Marionette>

6. Marionette Basics

[http://developer.vectorworks.net/index.php/Marionette\\_Basics](http://developer.vectorworks.net/index.php/Marionette_Basics)

7. Marionette ノードの作り方（詳解）

[http://developer.vectorworks.net/index.php/Marionette\\_Implement\\_a\\_Node](http://developer.vectorworks.net/index.php/Marionette_Implement_a_Node)

8. ノード一覧（作成中）

[http://developer.vectorworks.net/index.php/Marionette\\_Node\\_Reference](http://developer.vectorworks.net/index.php/Marionette_Node_Reference)

### お問い合わせ

エーアンドエー株式会社 研究開発室

〒101-0062 東京都千代田区神田駿河台 2-3-15

Tel. 03-3518-0554 / Fax 03-3518-0122

E-mail [develop@aanda.co.jp](mailto:develop@aanda.co.jp)

Web <http://www.aanda.co.jp/develop/>

平成28 年度 文部科学省委託事業実証講座  
社会基盤分野における建設IT 技術(BIM・CIM) に係る  
中核的専門人材養成プログラム開発プロジェクト

【建築BIM 講座】－コンピューターショナル・デザインとは－  
Vectorworks Architect で学ぶマリオネット ビジュアルプログラミング

2016年12月5日 初版  
著作・制作 エーアンドエー株式会社  
〒101-0062 東京都千代田区神田駿河台 2-3-15

- Vectorworksソフトウェアは、Vectorworks Inc.の登録商標です。
- その他すべての商標は、それぞれの権利帰属者の所有物です。
- 規格及び仕様予告なく変更されることがあります。
- 本書のデータの一部または全部の著作権法の定める範囲を超える複写、複製、転載、データファイル化することを禁じます。

2016 A&A Co.,Ltd. Printed in Japan.