BTS MEC – OUTILS DE PROGRAMMATION

Fiche n°3 – Récupérer les paramètres d’éléments, manipuler des listes et les exporter au format CSV

*Cette fiche fait partie d’un ensemble de documents destinés à faciliter l’appropriation des outils de programmation par les enseignants du BTS MEC suite à la réforme mise en œuvre depuis septembre 2021.*

*La modélisation sous forme d’objets est un élément essentiel du BIM. La maquette 3D n’est donc que la partie visible de la grande quantité d’informations organisée dans le modèle. Les outils de programmation sont donc souvent utilisés pour interroger ou de renseigner les données contenues dans le modèle.*

*Cette fiche présente la mise en œuvre d’un processus destiné à récupérer les paramètres d’éléments composant une maquette sous forme de listes puis à les exporter au format CSV.*

*Auteur :*

*Vincent Lefort - Lycée Cantau Anglet*

*vincent.lefort@ac-bordeaux.fr*

*Fichiers fournis :*

* *initiation\_dynamo.dyn*

***SOMMAIRE***

[1 Récupérer les propriétés d’objets composant une maquette à l’aide de Dynamo 2](#_Toc106253850)

[1.1 Sélectionner les objets concernés par la requête 2](#_Toc106253851)

[1.1.1 Solution 1 : en sélectionnant tous les éléments d’un type ou d’une catégorie donnée 2](#_Toc106253852)

[1.1.2 Solution 2 : en cliquant sur un élément en particulier dans le modèle 2](#_Toc106253853)

[1.2 Récupérer les valeurs des paramètres dans les objets sélectionnés 3](#_Toc106253854)

[1.3 Exporter une liste de valeur sous forme d’un fichier CSV 3](#_Toc106253855)

[1.4 Exemple 4](#_Toc106253856)

[2 Manipuler les listes 5](#_Toc106253857)

[2.1 Index de liste 5](#_Toc106253858)

[2.2 Générer des listes 5](#_Toc106253859)

[2.2.1 Utilisation de nœuds spécifiques 5](#_Toc106253860)

[2.2.2 Utilisation de nœuds « Code Block » 6](#_Toc106253861)

[2.3 Récupérer les valeurs contenues dans une liste 6](#_Toc106253862)

[2.4 Listes de listes 6](#_Toc106253863)

[2.5 Transposer des listes 7](#_Toc106253864)

[2.6 Autres manipulations de listes (trier, remplacer, etc.) 8](#_Toc106253865)

[3 Exemple 8](#_Toc106253866)

# Récupérer les propriétés d’objets composant une maquette à l’aide de Dynamo

Pour récupérer les propriétés des objets qui composent la maquette il faut procéder en plusieurs étapes :

1. Sélectionner les objets concernés par la requête,
2. Récupérer les valeurs des propriétés,
3. Exporter une liste de valeurs sous forme d'un fichier CSV.

**Remarque importante : Avec REVIT, quasiment toutes ces actions seraient réalisables à l’aide de nomenclatures, sans nécessiter de compétences en programmation !**

## Sélectionner les objets concernés par la requête

### Solution 1 : en sélectionnant tous les éléments d’un type ou d’une catégorie donnée

« All Elements of Type »

* Ce nœud se trouve dans la catégorie REVIT => SELECTION
* Il permet de récupérer une liste avec les identifiants des éléments d’un type donné
* Il est nécessaire de préciser en paramètre :
  + Le type des éléments concernés par la requête grâce au nœud « Element Types » (catégorie REVIT => SELECTION)

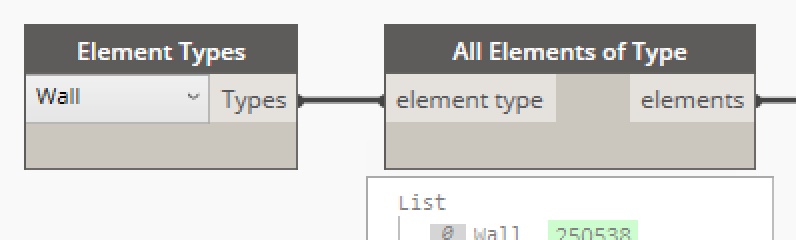


Figure - Mise en œuvre du nœud "All Elements of Type" afin de récupérer la liste des Murs contenus dans le modèle.

### Solution 2 : en cliquant sur un élément en particulier dans le modèle

« Select Model Element »

* Ce nœud se trouve dans la catégorie REVIT => SELECTION
* Il permet de récupérer l’identifiant de l’élément sélectionné
* Après avoir cliqué sur le bouton « Sélectionner », il suffit d’aller dans la fenêtre REVIT pour y sélectionner un objet.

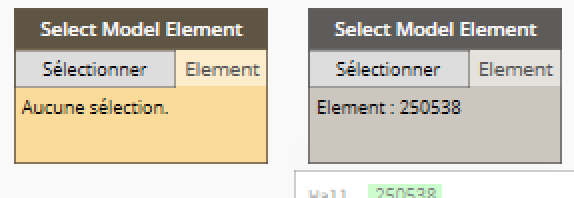


Figure - Mise en œuvre du nœud " Select Model Element", avant sélection et après sélection

## Récupérer les valeurs des paramètres dans les objets sélectionnés

« Get Parameter Value By Name »

* Ce nœud se trouve dans la catégorie REVIT => ELEMENTS
* Il permet de récupérer la valeur d’un paramètre d’un élément
* Il est nécessaire de préciser en paramètre :
  + Une liste avec les identifiants des éléments concernés par la requête
  + Une chaine de caractère (String) égale au nom du paramètre à récupérer

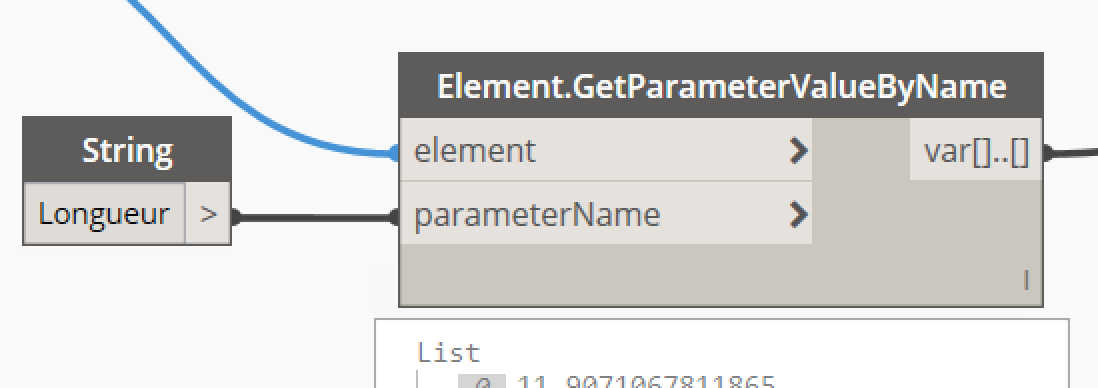


Figure - Mise en œuvre du nœud « Get Parameter Value By Name », ici sur un nœud avec un élément de type Mur.

Pour aider à déterminer le nom du paramètre à renseigner dans la commande « Get Parameter Value By Name », il est possible de visualiser l’ensemble des paramètres accessibles à partir du nœud « Element.Parameters ».

« Element.Parameters »

* Ce nœud se trouve dans la catégorie REVIT => ELEMENTS => ELEMENT
* Il permet d’afficher l’ensemble des paramètres associés à un élément

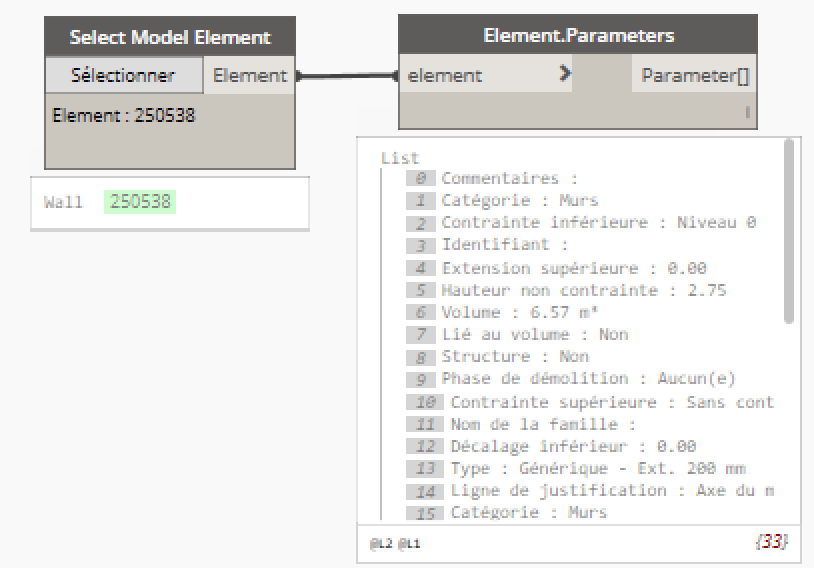
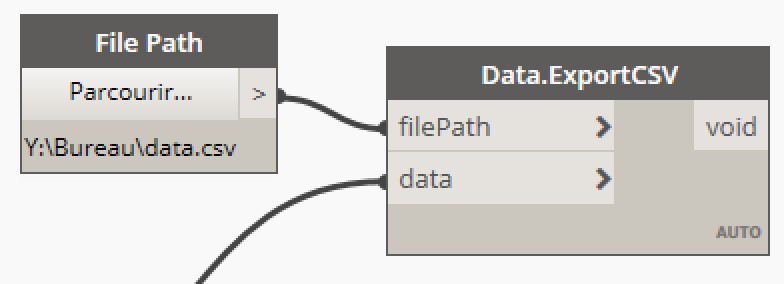


Figure - Exemple de résultats obtenu en exécutant le nœud "Element.Parameters" sur un élément de type Mur

## Exporter une liste de valeur sous forme d’un fichier CSV

« Data.ExportCSV »

* Ce nœud se trouve dans la catégorie IMPORTEXPORT => DATA
* Il permet d’exporter une liste de liste dans un fichier, les valeurs sont séparées par des virgules, les différentes listes sont écrites en lignes.
* Il est nécessaire de préciser en paramètre :
  + Le chemin du fichier d’export grâce au nœud « File Path » (catégorie IMPORTEXPORT => FILE SYSTEM)
  + Une liste de listes



## Exemple

Objectif : Exporter dans un fichier CSV la longueur des murs composant le modèle.

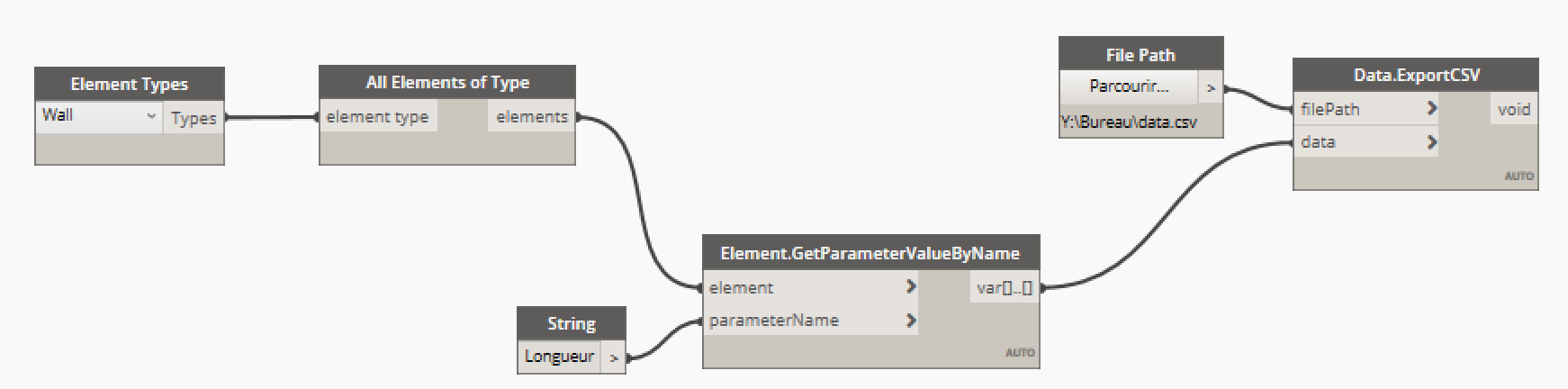


Figure - Vue d'ensemble du programme permettant d'exporter les longueurs des murs

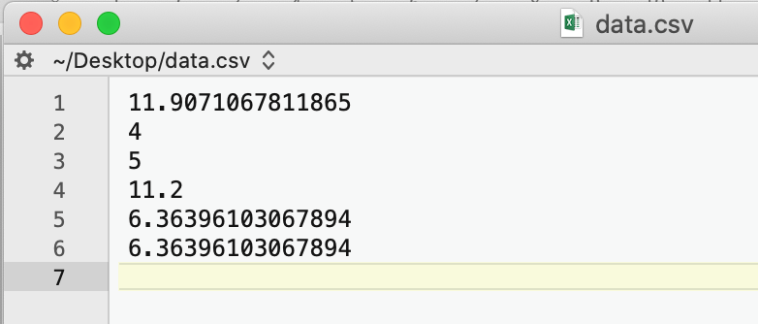


Figure - Fichier CSV généré par le programme exportant les longueurs des murs

# Manipuler les listes

*Partie largement inspirée du paragraphe 6. Conception avec des listes de la notice Dynamo (https://primer.dynamobim.org)*

Une liste est un ensemble d'éléments. Les listes sont souvent utilisées en programmation car il est plus rapide de manipuler un groupe d’éléments plutôt que chaque élément individuellement.

On peut manipuler une liste de nombre, mais aussi une liste d’objets, ou même une liste de listes.

## Index de liste

Par convention le premier élément d'une liste est toujours l’élément n°0 et non n°1. Lorsqu'il est question du premier élément d'une liste, il s'agit en fait de l'élément qui correspond à l'index 0.

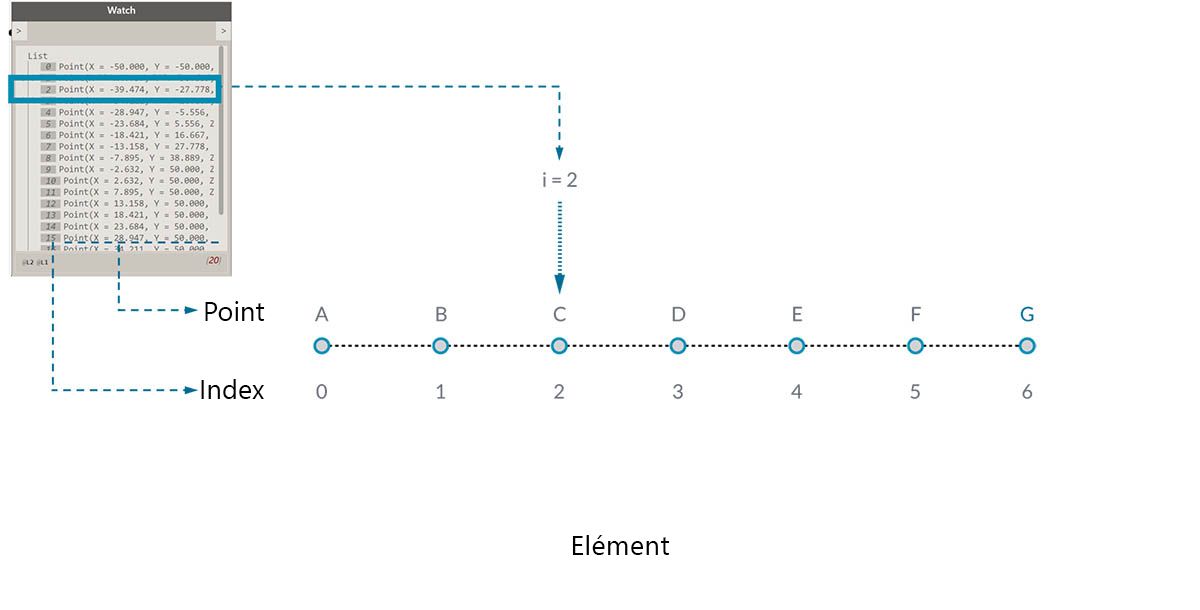


Figure - Exemple avec une liste de points (caractérisés par leurs coordonnées) affichée dans Dynamo. Le premier point est l'élément avec l'index 0.

## Générer des listes

### Utilisation de nœuds spécifiques

« Range »

* Ce nœud se trouve dans la catégorie LIST => GENERATE
* Il permet de créer une liste de nombre en spécifiant le nombre de départ, le nombre maximal et le pas.

« List Create »

* Ce nœud se trouve dans la catégorie LIST => GENERATE
* Il permet de créer une liste en indiquant explicitement les différents éléments composant la liste

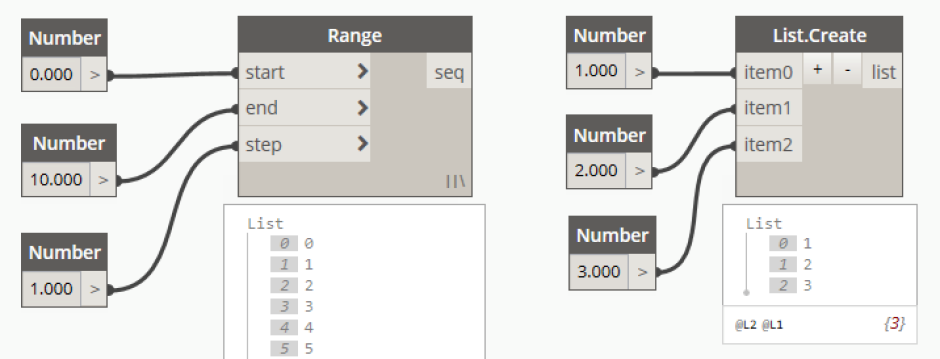


Figure - Mise en œuvre des nœuds Range et Liste.Create permettant de générer des listes

Les nœuds communément utilisés renvoient très fréquemment les résultats sous forme de liste à l’image de la liste des longueurs de murs générée plus haut.

### Utilisation de nœuds « Code Block »

Les nœuds « Code Block » sont accessibles dans la catégorie SCRIPT => EDITOR

La commande 0..10 ; génère une liste allant de 0 à 10, par pas de 1.

La commande 0..10#10 ; génère une liste allant de 0 à 10 en 10 pas.

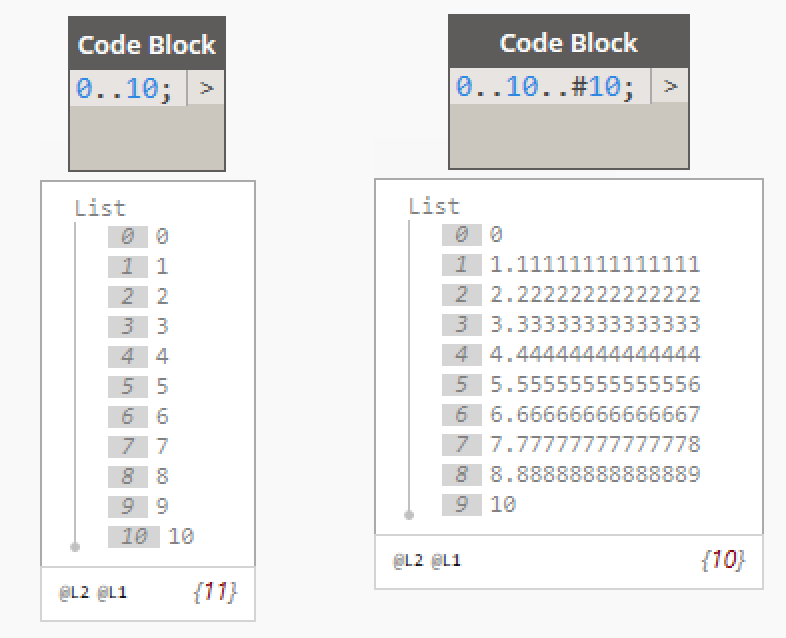


Figure - Génération de listes à partir du nœud Code Block

## Récupérer les valeurs contenues dans une liste

Pour récupérer une valeur contenue dans une liste, il est possible d’utiliser le nœud « List.GetItemAtIndex »

« List.GetItemAtIndex »

* Ce nœud se trouve dans la catégorie LIST => INSPECT
* Il permet de récupérer la valeur stockée dans une liste à un index donné

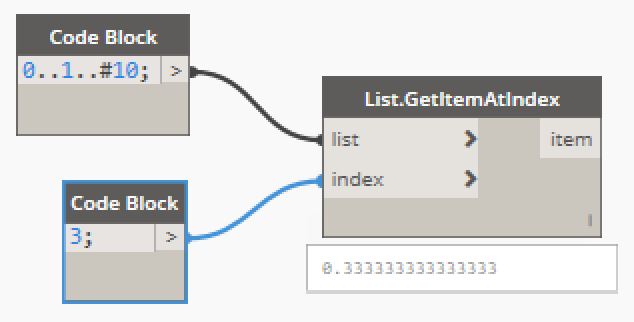


Figure - Récupération de la 4ème valeur (index = 3) d'une liste à l'aide du nœud "List.GetItemAtIndex"

## Listes de listes

Il est possible de générer des listes à partir du nœud « List Create ». Par exemple dans l’exemple ci-dessous nous générons une liste avec les nombres premiers, et une liste avec les nombres de Fibonacci.

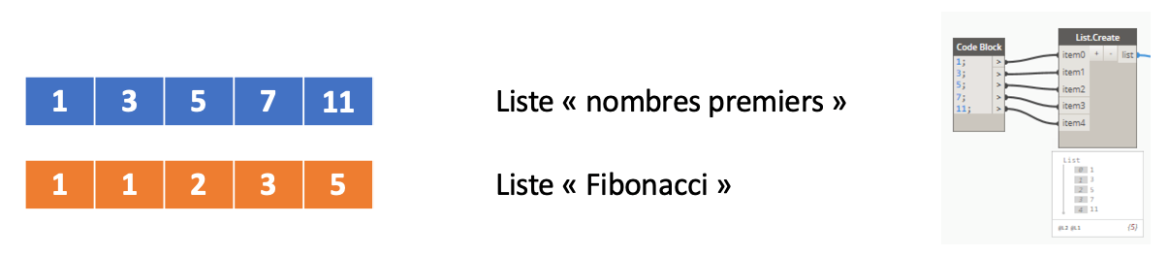


Figure - Génération de plusieurs listes de nombres à l'aide du nœud « List.Create »

Mais il est également possible de générer des listes de listes à partir de ce même nœud.

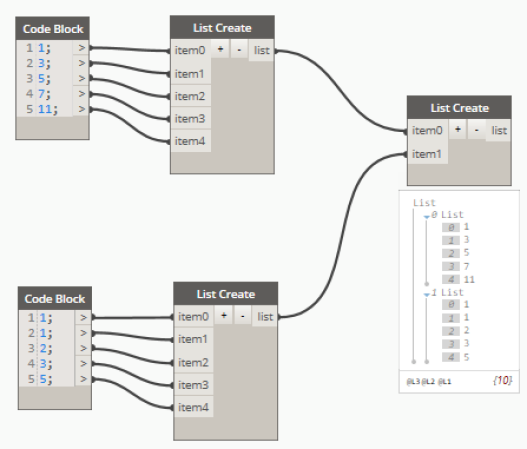
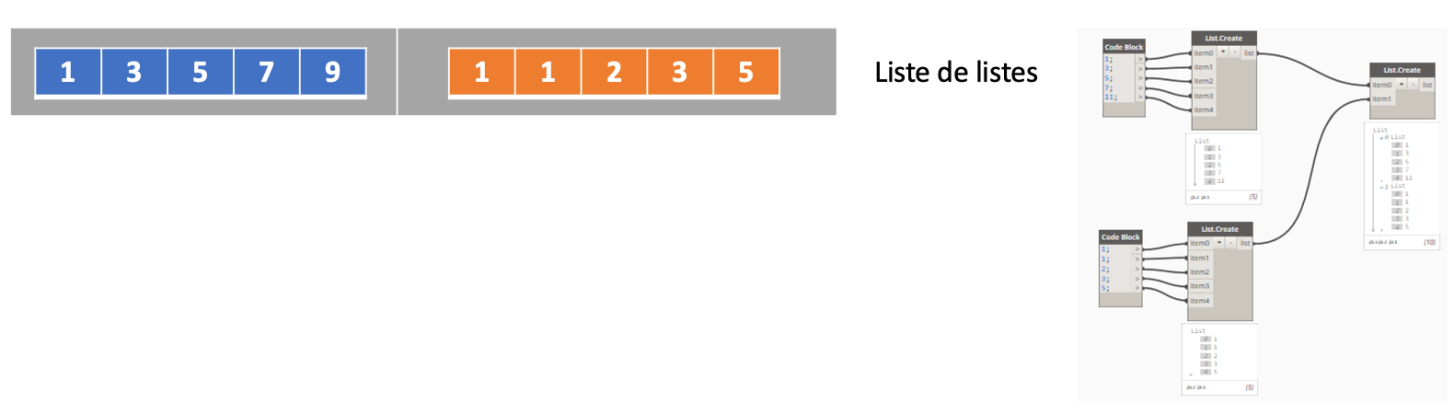


Figure - Génération de listes de liste à partir du nœud « List Create »

## Transposer des listes

La transposition des listes est une action fréquente notamment lors des import/export de données. En effet les données sont par défaut écrites/lues en lignes successives par Dynamo (et de manière générale en programmation), or nous avons souvent l’habitude d’organiser nos données en colonnes lorsque nous faisons des tableaux.

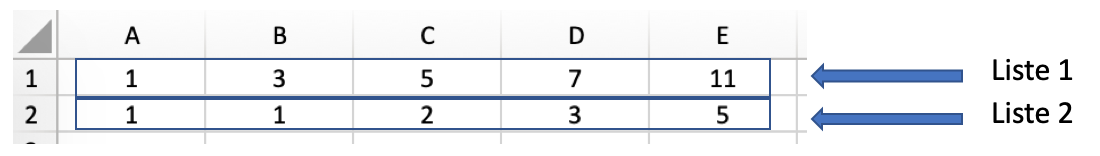


Figure - Stockage d'une liste de listes de nombres représentée avec un tableur : chaque liste de nombre est organisée sur une ligne

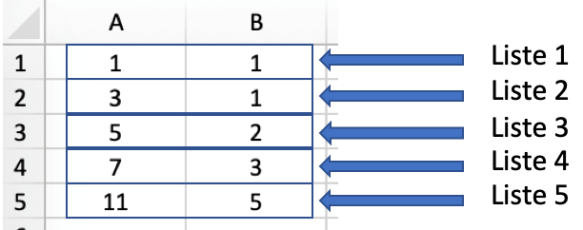


Figure - Stockage d'une liste de listes de nombre après transposition (figure faite avec un tableur)

« List.Transpose »

* Ce nœud se trouve dans la catégorie LIST => ORGANIZE
* Il permet de transposer une liste de listes

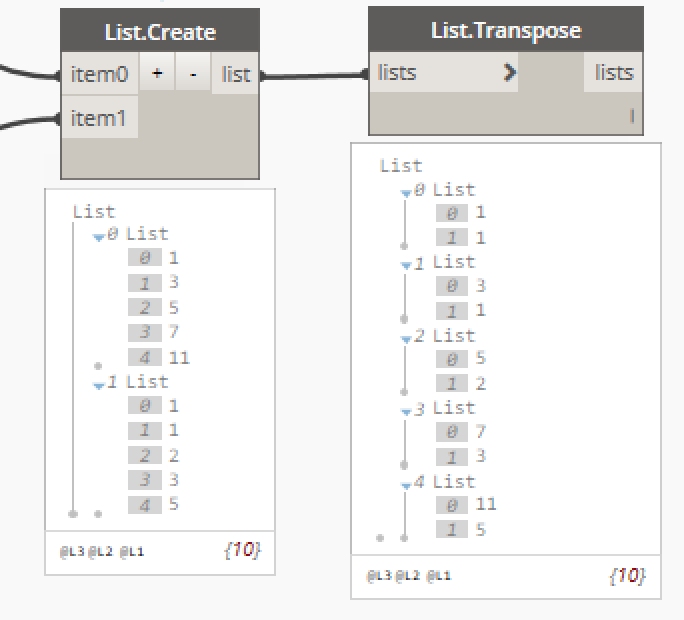


Figure 15 - Transposition de listes à l’aide du nœud « List.Transpose »

## Autres manipulations de listes (trier, remplacer, etc.)

De nombreux nœuds permettent d’agir sur les listes (trier, remplacer, transposer, etc.), où d’en extraire des caractéristiques (longueur, valeur pour une index donné, valeur maximale ou minimale, etc.).

Ces nœuds sont accessibles dans la catégorie LIST.

# Exemple

**Objectif :** A partir des nœuds vus précédemment, créer une liste numérotée des vues de la maquette dans un fichier CSV semblable à celui présenté ci-dessous.

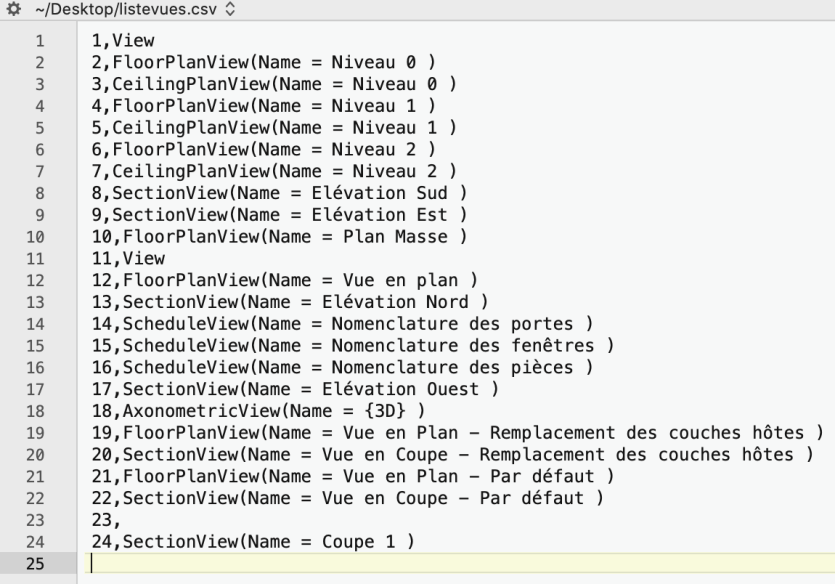


Figure 16 – Fichiers CSV à créer à partir de la maquette

**Démarche à suivre :**

* Créer une liste de tous les éléments de type vue
* Créer une liste de numéros allant de 1 jusqu’au nombre de vues afin de numéroter les vues
* Créer une liste globale rassemblant la liste de numéros, et la liste des vues
* Transposer la liste de liste pour organiser les données en colonnes
* Exporter la liste au format CSV

**Solution proposée :**

