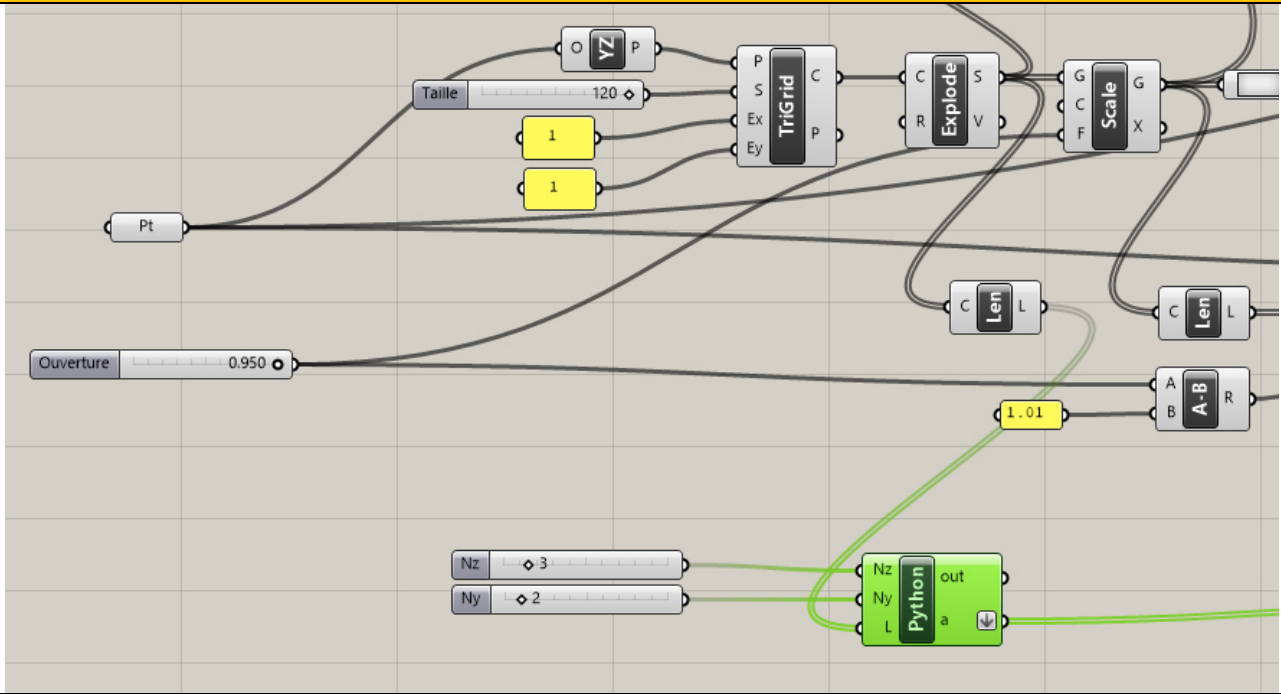
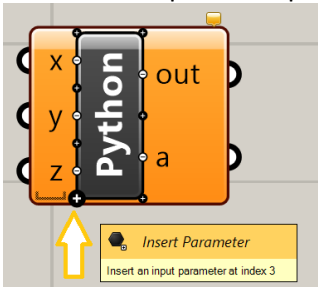
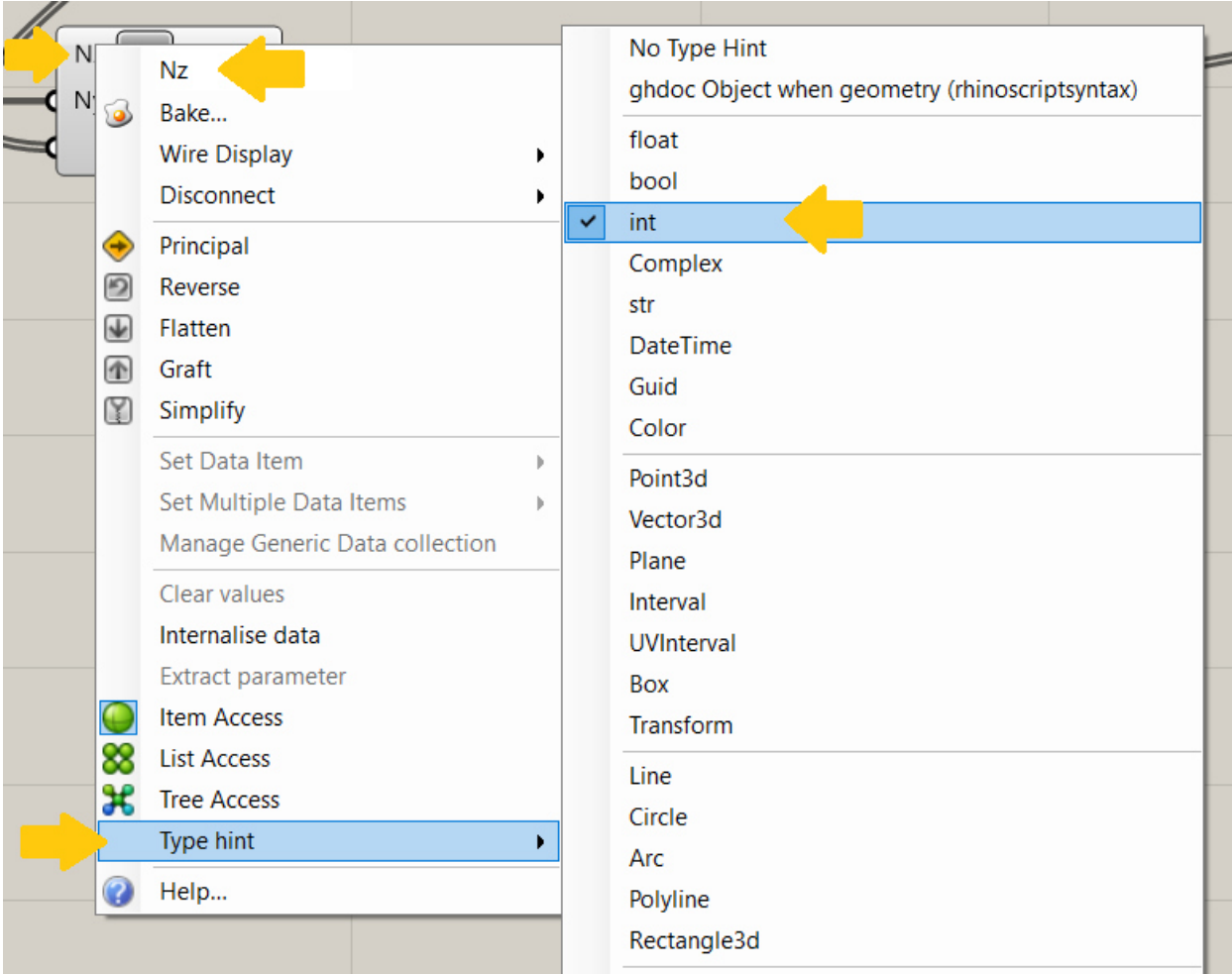




Compléter l’algorithme Gh pour réaliser un pavage des modules dynamiques

Commandes et liens entre les blocs GH :	
	<p>length Renvoie la longueur L des cotés des triangles équilatéraux</p>
	<p>0..10 Faire deux numbers sliders</p>
	<p>GhPython Script Permet d’accéder à l’éditeur Python</p> <p>Zoomer et cliquer sur + pour ajouter une variable</p> 
Définition des variables :	
	<p>sur x :</p> <p>1- Renommer en Nz</p> <p>2- Définir le type de variable, en cliquant sur Type hint, puis cocher int (Nombre entier)</p>
	<p>sur y :</p> <p>1- Renommer en Ny</p> <p>2- Définir le type de variable, en cliquant sur Type hint, puis cocher int (Nombre entier)</p>
	<p>sur z :</p> <p>1- Renommer en L</p> <p>2- Définir le type de variable, en cliquant sur Type hint, puis cocher float (Nombre à virgule)</p>
Taper le programme en python (page suivante)	

Grasshopper Python Script Editor

File Edit Tools Mode Help

Test OK

```
1  """Provides a scripting component.
2  ....Inputs:
3  .....Ny: nombre entier de module en y
4  .....Nz: nombre entier de module en z
5  .....L: longueur (nombre à virgule) d'un module
6  ....Output:
7  .....a: point avec coordonnées x,y,z"""
8
9
10 import rhinoscriptsyntax as rs
11 import math
12
13 a=[]
14 z=()
15 y=()
16 j=0
17 i=0
18 while (j<Nz):
19     j += +1
20     if (j%2==0):
21         y= -3*L
22         for i in range(Ny):
23             y += +3*L
24             z = j*L/2/math.tan(math.pi/6)
25             p=rs.AddPoint(0,y,z)
26             a.append(p)
27     else:
28         y=-4.5*L
29         for i in range(Ny):
30             y += +3*L
31             z = j*L/2/math.tan(math.pi/6)
32             p=rs.AddPoint(0,y,z)
33             a.append(p)
```

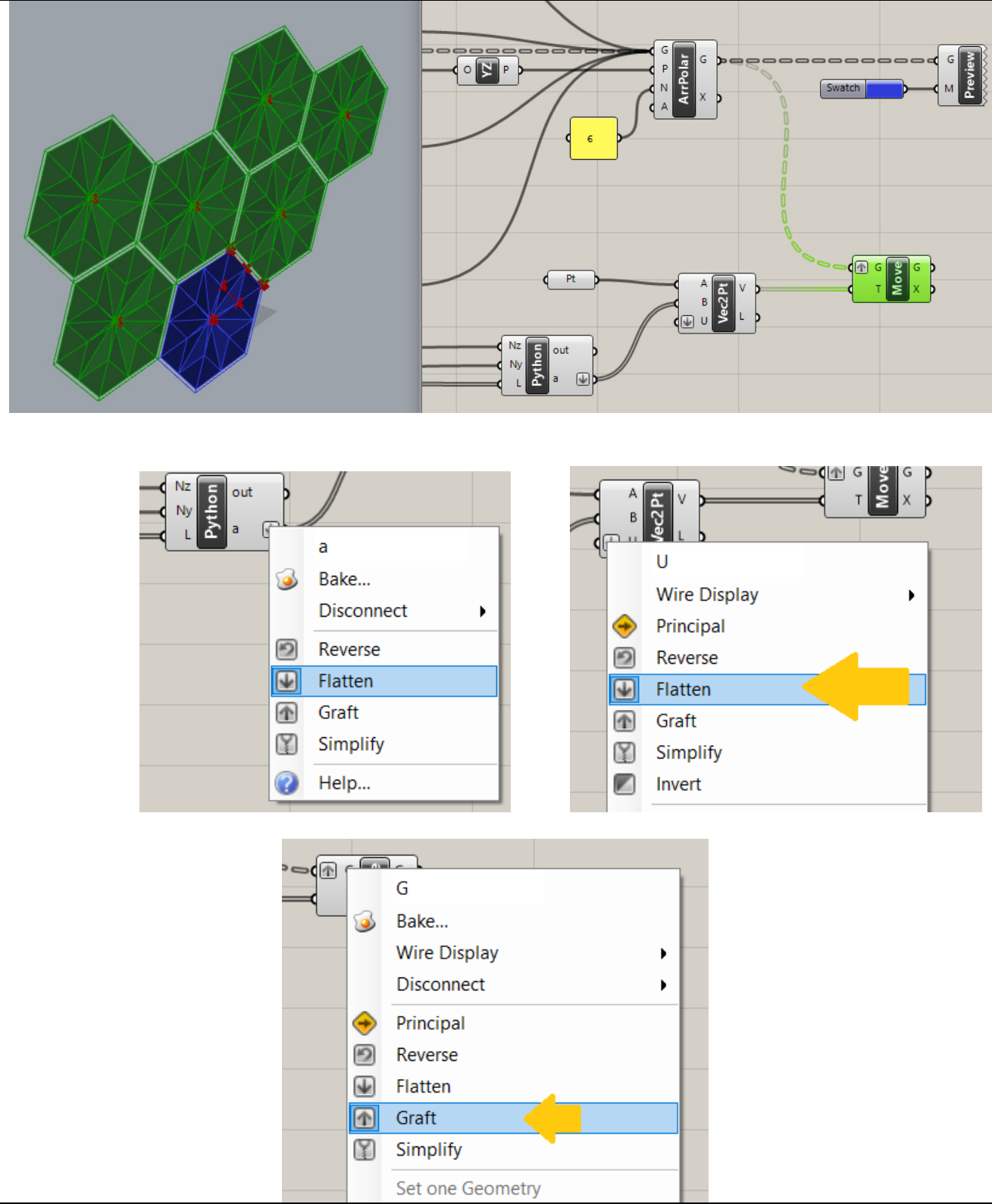
Double clic sur le bloc Python pour **ouvrir l'éditeur**.

Remarque :

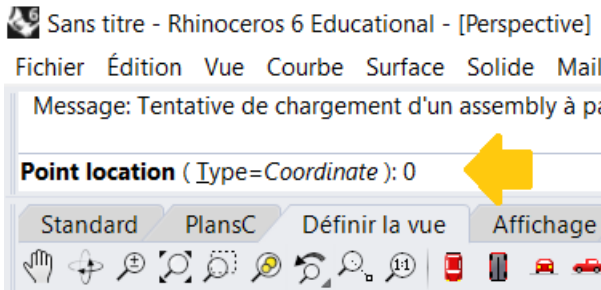
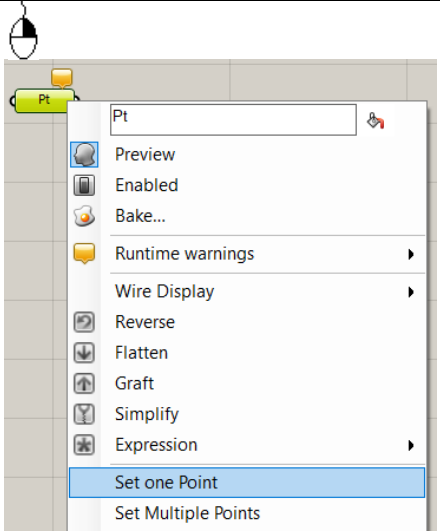
La touche **Tabulation** permet d'aligner correctement les commandes python

La commande ► **Test** permet de tester votre programme et mettre en évidence vos erreurs de syntaxe.

Continuer l'algorithme :



Point



Sur le bloc Python

sur **a** et sélectionner **Flatten** – une petite **flèche vers le bas** apparait à coté de a
⇐ voir ci-contre

Vector 2pt

sur **u** et sélectionner **Flatten** – une petite **flèche vers le bas** apparait à coté de u
⇐ voir ci-contre

Move

sur **G** et sélectionner **Graft** – une petite **flèche vers le haut** apparait à coté de G
⇐ voir ci-contre