

# Python et programmation par blocs en mathématiques avec SofusPy

[patrick.raffinat@univ-pau.fr](mailto:patrick.raffinat@univ-pau.fr)

## A) Introduction

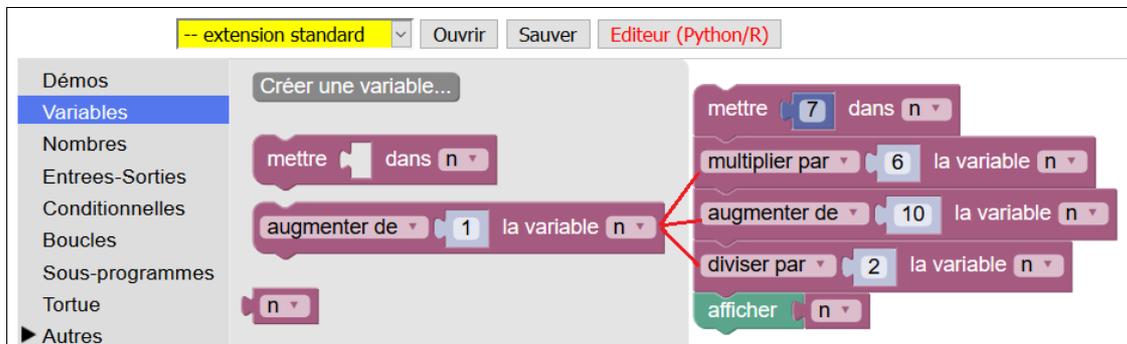
Dans le premier parcours d'été de 2017, j'avais souligné [l'intérêt de passerelles en algorithmique](#), en particulier pour passer de la programmation visuelle (au collège) à Python (au lycée) grâce au logiciel SofusPy qui venait d'être créé. Plusieurs articles lui ont été consacrés lors de l'année scolaire 2017-2018 :

- N° 57 : [De la programmation par blocs à Python, avec SofusPy et PluriAlgo](#)
- N° 58 : [Les listes en Python et en programmation visuelle avec SofusPy](#)
- N° 60 : [Scratch ou Blockly au collège : un nouveau leurre ?](#)

Comme les fonctionnalités ont continué à évoluer, un article de synthèse m'a semblé nécessaire et sera publié dans le N°63 (janvier 2019). En attendant, en voici un extrait dont le but est de faciliter dès à présent le travail d'enseignants qui envisageraient d'utiliser SofusPy en 2018-2019. SofusPy est disponible en ligne sur mes pages personnelles consacrées à Blockly : <http://raffinat.perso.univ-pau.fr/blockly/index.html> .

## B) Création et transformations de variables

Grâce à un changement de version de Blockly, il y a désormais dans le menu Variables un bouton « Créer une variable » qui rend plus naturelle cette tâche.



Ce bouton fait aussi apparaître dans le menu Variables un bloc permettant de transformer des variables, avec diverses options : augmenter, diminuer, multiplier, diviser, augmenter d'un pourcentage... Pour y parvenir, une adaptation du Blockly de base a été réalisée car la seule transformation proposée était « augmenter ».

## C) Trois configurations : standard, calcul formel ... et R !

Deux configurations (calcul symbolique et R) ont été ajoutées à la version initiale de SofusPy. Elles sont accessibles grâce à une liste déroulante :

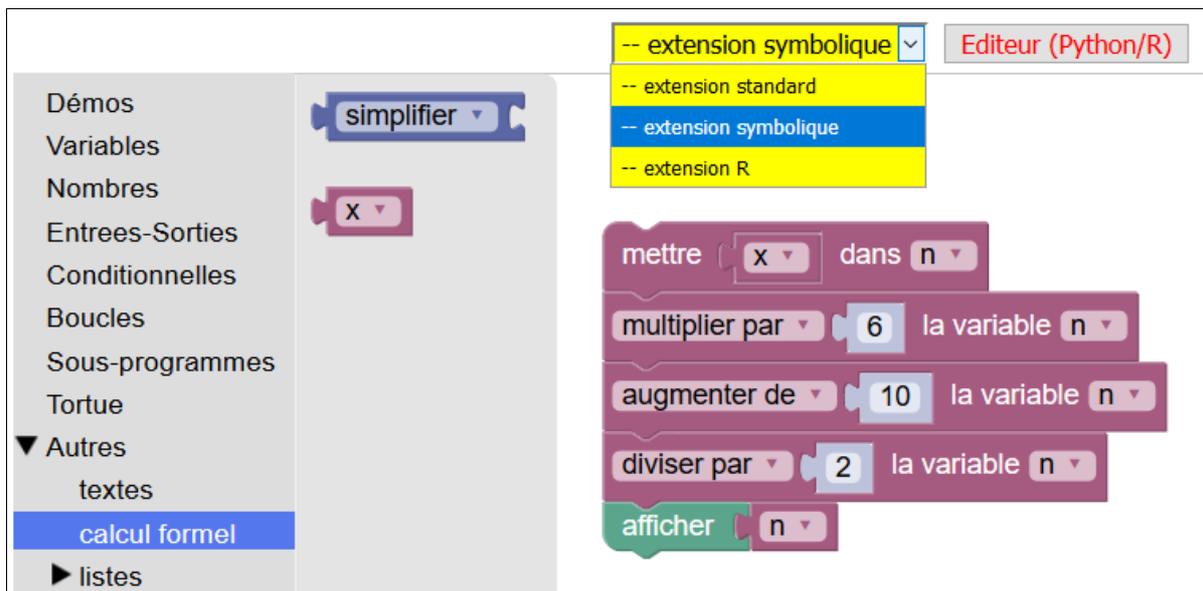


## Configuration standard

La configuration standard, proposée par défaut, était la seule disponible dans la version 2017 de SofusPy. Elle permet de passer d'un environnement de programmation par blocs à un environnement Python (bouton « Editeur ») ou l'inverse (bouton « retour à Blockly »). Le bouton « Editeur » effectue également la traduction en Python du programme visuel courant, afin de faciliter la transition entre programmation par blocs (au collège) et Python (au lycée).

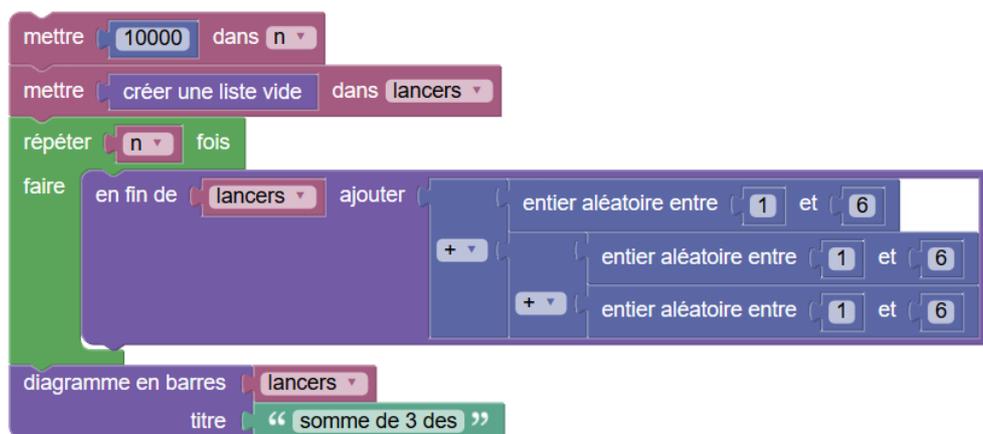
## Configuration formelle

Cette configuration permet de faire du calcul symbolique, que ce soit en mode visuel ou en mode Python (voir <http://revue.sesamath.net/spip.php?article1076>) :



Pour pouvoir être exécuté, un programme par blocs est traduit en Python. Comme l'interpréteur de SofusPy ne gère pas le module de calcul formel de Python (SymPy), le code engendré est alors envoyé à un serveur externe (SymPy Live).

## Configuration ... R !



En configuration standard, on peut seulement récupérer le code Python (bouton Editeur) pour le transférer et l'exécuter dans un environnement externe, à cause du graphique. C'est certes appréciable, mais une exécution directe du programme par blocs apporterait un plus non négligeable : c'est ce que permet la configuration R.

## D) Activités

Dans le N°63, je fournirai le code source de SofusPy pour qu'il puisse être installé localement et, en configuration standard, fonctionner sans connexion à Internet. Une autre motivation pour mettre ce code source à disposition est d'encourager la création d'activités (stages de formation ou utilisation en classe). Cet article montrera comment elles peuvent être (assez facilement) intégrées au menu de SofusPy.

Je poursuivrai en présentant quatre activités actuellement disponibles. Leurs thèmes (pourcentages, sommes, produits et dés) reflètent des problématiques que je n'ai pas cherché à relier précisément aux programmes officiels. Elles sont à adapter aux objectifs pédagogiques d'une classe donnée et aux connaissances (mathématiques et algorithmiques) des élèves.

Je les commenterai dans le N°63, ce qui n'empêche pas de les tester dès à présent... Et, il n'y aura pas à attendre le N°63 pour lire <http://revue.sesamath.net/spip.php?article1098> puisque MathémaTICE a une rubrique « un avant-goût des sommaires à venir »...

## E) SofusPy comme composant de PluriAlgo

SofusPy peut être utilisé soit séparément, soit comme composant du logiciel PluriAlgo ([voir N°57](#)). Des formulaires de PluriAlgo facilitent le codage (entrées-sorties, sous-programmes, sommation, comptage...) dans différents langages textuels dont Python ou, nouveauté décrite dans le N°63, en programmation par blocs :

The image shows two side-by-side panels of the SofusPy interface. The left panel is in text-based editing mode, showing input fields for 'entrees' (a b) and 'sorties' (div), a 'calculs' dropdown set to 'fonctions', and a 'Nouveau' button. Below these are blocks of code: 'demander " a ? " et mettre dans a', 'demander " b ? " et mettre dans b', 'mettre pgcd a b dans div', 'afficher div', and a function definition 'definir pgcd avec : a, b' that returns a value. The right panel is in block-based editing mode, showing a 'pour' loop from k=1 to 1000, a 'comptage de' checkbox checked, and a condition '== 6'. A 'Créer' button is visible. The block-based code includes 'mettre 0 dans nb\_6', a 'pour k de 1 à 1000 par 1' loop, a 'faire' block with a 'si == 6' condition containing 'mettre nb\_6 + 1 dans nb\_6', and 'afficher nb\_6'.

La nature du code produit (textuel ou visuel) dépend de l'onglet d'édition sélectionné avant de cliquer sur le bouton « Nouveau » ou sur le bouton « Créer ».