

## L'intérêt de passerelles en algorithmique



Dans son [manuel de cycle 4](#), Sésamath a illustré le chapitre consacré au codage avec un langage visuel (Scratch) et un langage textuel (Python). Mais, comme c'est souvent le cas quand on parle de codage dans l'enseignement secondaire, les modalités de la transition entre programmation visuelle et programmation textuelle ne sont pas évoquées, ce qui laisse à penser qu'elle est uniquement de nature chronologique : on initie d'abord les collégiens de façon ludique avec Scratch, puis on passe à autre chose au lycée (ou en fin de collège) avec la programmation textuelle. La continuité pédagogique est alors réduite à peau de chagrin : il y a certes des boucles (des instructions conditionnelles...) dans les deux cas, mais ça s'arrête là.

Il est à craindre qu'une telle approche n'entraîne une attitude de rejet de la part de nombreux élèves lorsqu'ils aborderont la programmation textuelle : non seulement la programmation textuelle demande plus d'efforts, mais les applications développées sont moins valorisantes dans un premier temps car elle requiert plus de compétences. En d'autres termes, **l'avancée pédagogique que constitue la programmation visuelle risque d'être anéantie si on ne fait pas l'effort de rendre plus attractif l'enseignement des langages textuels.**

Pour y parvenir, une piste à privilégier est l'existence d'une **passerelle** entre Blockly et deux langages textuels populaires (Python et Javascript <sup>1</sup>), passerelle qui peut être étendue vers des logiciels mathématiques (Xcas, R) :

- [SofusPy, une passerelle entre programmation visuelle et Python](#)
- [Programmer par blocs au lycée et en IUT](#)
- [Mathém@ALGO, une suite logicielle très polyvalente](#)
- [Blockly, calcul formel et statistiques](#)

Il faudrait aussi cesser de croire qu'il est aisé pour tous les élèves de coder dans un langage de

<sup>1</sup> Par exemple, quand on réussit à résoudre une des étapes de jeux (labyrinthe, tortue... ) Blockly ( voir <https://blockly-games.appspot.com/> ), la récompense qu'on reçoit fièrement est un code JavaScript.

programmation. Les difficultés ne sont pas seulement d'ordre syntaxique, mais aussi d'ordre sémantique : faut-il par exemple vraiment s'étonner que des élèves ne trouvent pas naturel de traduire un « pour k allant de 1 à 4 » en « for k in range(1:5) » (Python) ou « for(k=1 ; k<=4 ; k++ ) » (Javascript) ? Il est donc important de laisser le temps aux élèves d'assimiler les spécificités sémantiques d'un langage et de leur proposer un environnement de programmation convivial facilitant le codage.

Il y a même des environnements multilingues proposant une **passerelle** entre langages de programmation, ce qui peut se révéler utile pour des élèves abordant plusieurs langages au cours de leur scolarité :

- Proglab (voir site <http://proglab.fr/> et l'article <http://www.apmep.fr/Proglab-l-algorithmique-accessible>) est un environnement en ligne permettant de programmer en AlgoBox ou en Javascript, et proposant un traducteur bidirectionnel entre ces 2 langages.
- PluriAlgo est un logiciel facilitant l'écriture d'algorithmes dans plusieurs langages et proposant une aide à la traduction : voir « [Version en ligne du logiciel d'algorithmique PluriAlgo](#) » (N°44)

L'offre en environnements conviviaux est particulièrement riche pour Python, l'option ISN n'y étant évidemment pas étrangère. Le logiciel EduPython (<http://edupython.tuxfamily.org/>) combine plusieurs atouts : nombreux packages et outils, documentation adaptée à l'enseignement des mathématiques (exemple <http://download.tuxfamily.org/edupython/EduPython1.3.pdf>)...

Quelle que soit la voie choisie pour enseigner la programmation textuelle, elle doit être à la hauteur de l'avancée pédagogique que constitue la programmation visuelle. Pour encourager cette avancée dans un cadre mathématique, l'IREM de la Réunion a décidé d'héberger plusieurs extensions de Blockly mentionnées dans ce parcours d'été (voir <http://irem.univ-reunion.fr/blockly/extensions/>).