



## Synthèse des TraAM 2017-18

publié le 14/06/2018 - mis à jour le 03/01/2019

### Descriptif :

Qu'a pu retirer le groupe de travail de l'académie de Poitiers de ces travaux académiques mutualisés ? Quelques résultats-clés sur la continuité des apprentissages de la 3ème à la 2nde pour l'algorithmique et la programmation.

Six conclusions ont été tirées des [travaux académiques mutualisés 2017-18](#).

Pour une appropriation interactive de celles-ci, le groupe propose au lecteur un diaporama qui permet la réalisation d'un **débat mobile** à organiser avec un groupe de collègues :

 [synthese\\_traam\\_2017-18\\_debat\\_mobile](#) (PDF de 966 ko)

Voici les six conclusions les plus importantes retenues par le groupe :

- **Développer la pensée algorithmique passe par des temps de métacognition : activités débranchées, analyse d'algorithmes, formalisation des notions : limites du « Jourdainisme ».**

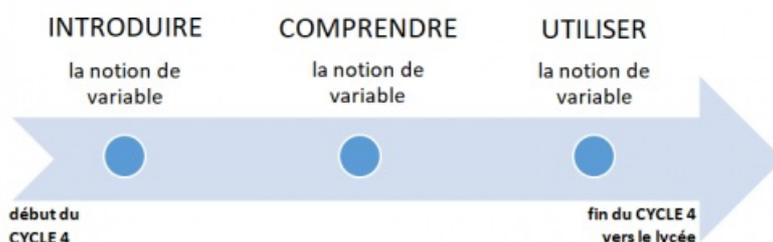
Nous avons pu nous apercevoir au collège, qu'à l'issue d'une succession d'activités avec Scratch, certaines maîtrises expertes atteintes par des élèves ne témoignaient néanmoins pas d'une compréhension profonde et explicite des notions algorithmiques (notions de boucle, de test...). Sans être exclusif ni systématique, le passage par des activités débranchées comme l'[exemple de la tortue](#) peut permettre de mieux intégrer et formaliser ces notions.

- **Cela n'empêche pas les démarches essais-erreurs qui restent motivantes, efficaces - même en Python**

Un des attraits qu'a la programmation pour des élèves en cours de mathématiques, c'est le statut de l'erreur qui est, par nature, différent de celui perçu par les élèves en maths. L'erreur ici est visible, elle permet de mieux comprendre le programme, elle développe l'esprit critique et l'analyse de problèmes. La nature de Scratch permet un retour (*feedback*) immédiat sur les choix qui ont été fait. Ce bénéfice se garde dans le passage à Python en Seconde. L'activité de [la boîte de conserve](#) permet aux élèves de déterminer lequel de deux programmes Python va réaliser la tâche demandée. Les élèves peuvent utiliser la fonction "Print" pour tester et ainsi mieux comprendre leur programme.

- **La notion de variable est à construire explicitement (activités dédiées)**

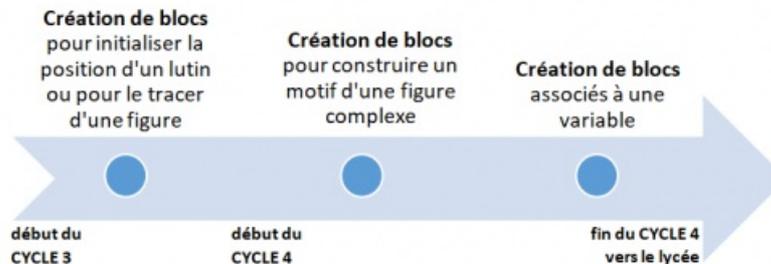
L'article [Progressivité sur la notion de variable](#) propose d'introduire la notion de variable par des activités dédiés, afin de mieux la comprendre et ainsi de mieux l'utiliser. En effet, certaines séances d'expérimentations effectuées par le groupe ont montré des élèves capable d'utiliser le bloc "Réponse" de Scratch, voire même de créer des variables, sans être capable d'expliquer (donc de comprendre profondément) cette qualité de boite-dont-la-valeur-peut-changer-au-cours-du-programme des variables.



- **La notion de fonction peut être pensée dès le collège par la création de blocs Scratch**

L'article [Progressivité sur la notion de fonction](#) évoque bien cette notion de fonction informatique. Vue au collège sous la forme d'un bloc de Scratch, elle n'a pas en particulier nécessairement de variable.

La qualité que possède un programme à pouvoir s'intégrer comment sous-programme d'un autre programme se traduit dans Scratch par la création de bloc. On peut ainsi concevoir un bloc "motif" qui va permettre le tracé d'un motif géométrique. Ce bloc est ainsi une nouvelle fonction (informatique) que l'on pourra réutiliser à divers endroits d'un programme de constructions graphiques.



Au lycée, cette même qualité d'intégrabilité dans un autre programme se traduit par la création de fonction de façon plus explicite comme par exemple dans l'exemple de [la boule du Futuroscope](#).

- **La notion de fonction questionne la définition Entrées-Instructions-Sorties de l'algorithme et place les instructions Input/Print au rang d'outils de débogage.**

Le groupe a pu réaliser qu'il existe des algorithmes sans entrée (générations de nombres aléatoires par exemple) ou sans sortie (comme classer des nombres dans l'ordre croissant : à l'issue de l'algorithme on a bien une action qui a été réalisée - les nombres sont dans l'ordre croissant - mais sans nécessairement demander la suite composée de ces nombres). Par ailleurs, dès que l'on demande une entrée avec "Input", le programme nécessite une action de l'utilisateur et ne peut plus être intégré dans un programme plus grand. Cela a beaucoup fait évoluer les représentations, des collègues de lycée notamment. Il a été intéressant pour cela de comparer l'activité [géométrie repérée et algorithmique](#) et celle de [la boîte de conserve](#).

La notion de fonction informatique a également été l'occasion de travailler celle de fonction mathématique, au cœur du programme de seconde.

- **L'activité algorithmique au service de la résolution de problèmes mathématiques doit être pensée avec la double complexité des tâches qu'elle engendre (modèle construit par l'équipe).**

D'après les programmes, l'activité algorithmique au lycée doit être au service de la résolution de problèmes mathématiques. Or, comme le montre le schéma suivant, qui combine le travail de création d'une taxonomie sur les tâches algorithmiques et celui de Marc-André Lalande du groupe de recherche RECIT au Québec, il peut être contre-productif de demander une tâche algorithmique trop haute dans la taxonomie sur un thème mathématique éloigné de sa Zone Proximale de Développement (ZPD). Pour faire simple, demander à un élève de créer, de toute pièce, un algorithme pour résoudre un problème statistique, nécessite que ce dernier soit familier de la notion de fréquence. Dans le cas contraire, les deux difficultés, algorithmique et mathématique, vont se conjuguer pour un risque de décrochage important. Il sera alors préférable de fournir tout ou partie de l'algorithme en demandant à l'élève de chercher à comprendre ce qu'il fait, ou de le compléter.

A noter que tous les élèves n'ont ni la même ZPD, ni les mêmes compétences en programmation. Il a pu être ainsi intéressant de proposer des activités différenciées comme on peut en trouver dans le [livret Scratch](#) de l'académie.

# Résoudre un problème mathématique par l'algorithmique et la programmation

Apprentissage combiné au lycée

