



Faciliter la transition collège-lycée en algorithmique et programmation

publié le 01/12/2024 - mis à jour le 09/12/2024

Descriptif :

Une première séance diagnostique en algorithmique et programmation qui s'adresse aux enseignants de mathématiques de seconde générale et de troisième. Cette séance s'appuie sur la ressource numérique Citizen Code Python.

Sommaire :

- Contexte et objectif de la séance
- Plus-value du numérique dans cette séance
- Modalités de mise en œuvre
- Déroulement de la séance
- Compétences numériques travaillées
- Ressources et documents
- Bilan critique de la séance
- Conclusion

● Contexte et objectif de la séance

Cet article présente une séance qui s'adresse aux enseignants de mathématiques de seconde générale et de troisième. Cette séance s'appuie sur la ressource numérique Citizen Code Python pour répondre à plusieurs objectifs pédagogiques :

- Évaluer de façon diagnostique les compétences des élèves en algorithmique et en programmation.
- Évaluer les représentations que les élèves se font de cette discipline.
- Faire évoluer ces représentations de manière positive, en renforçant leur confiance dans leur capacité à programmer, particulièrement chez les élèves qui doutent de leurs compétences, et en portant une attention (discrète mais) particulière aux filles.

Pour accompagner cette transition entre le collège (programmation événementielle, par blocs) et le lycée (programmation impérative textuelle), la ressource Citizen Code Python offre un parcours exploratoire permettant à chaque élève de tester ses compétences dans un environnement numérique ludique et motivant. Ce parcours facilite ainsi l'entrée dans l'algorithmique et la programmation au lycée en prenant appui sur les acquis de collège.



● Plus-value du numérique dans cette séance

Si le numérique n'est pas toujours requis pour travailler l'algorithmique et la programmation, ici, l'évaluation diagnostique des compétences algorithmiques des élèves dans un contexte "réel" de programmation nécessite à mon sens l'utilisation d'un environnement numérique. La ressource Citizen Code Python permet ainsi de mettre les élèves en situation directe de programmation sur ordinateur ou tablette.

Pour les élèves, la ressource :

- offre un cadre ludique et motivant qui leur permet de tester leurs compétences tout en réalisant des missions ;
- favorise leur confiance en leur capacité de réussite en algorithmique et programmation, en valorisant la progression individuelle, le feedback positif, le droit à l'erreur, et l'expérience d'un effort récompensé.

Pour l'enseignant :

- elle fournit un cadre unifié pour prendre la mesure des compétences des élèves dans un espace centralisé ;
- elle permet d'assigner des missions spécifiques pour évaluer des compétences précises (ex. : instructions, boucles, conditions) ;
- elle permet un suivi en temps réel des progrès et difficultés rencontrés par les élèves ;
- elle s'adapte aux rythmes et besoins des élèves, grâce à la possibilité de travailler en Blockly (équivalent de Scratch) ou en Python, ce qui offre un accompagnement différencié.



La ressource est accessible via le GAR, permettant un accès intégré depuis l'ENT, simplifiant ainsi l'accès pour les enseignants et les élèves, une fois l'activation faite par le RUPN de l'établissement.

● Modalités de mise en œuvre

- **Niveau :**
Seconde pour évaluer les acquis en programmation au collège ; 3e pour renforcer les bases en programmation par blocs et préparer l'entrée dans le langage Python en vue du lycée.
- **Durée :** 50 minutes
- **Ressources numériques utilisées :** Citizen Code Python ([lien direct](#)) mais accès via le Médiacentre de l'ENT) ou alternativement, la plateforme [Algorea](#), questionnaires pré-test et post-test sur les représentations (format papier possible).
- **Prérequis notionnels :** Aucun. Le parcours introduit à l'algorithmique et à la programmation en Seconde en s'appuyant sur les connaissances de collège (instructions, boucles, conditions). Des fiches d'accompagnement pour chaque mission sont téléchargeables.
- **Prérequis techniques :**
 - Les élèves doivent avoir accès à un ordinateur (ou une tablette) avec connexion pour un suivi

individualisé.

- L'enseignant doit activer Citizen Code Python via le Médiacentre, avec assignation du parcours exploratoire.
- Les élèves se connectent une première fois avant la séance pour apparaître dans la classe de l'enseignant.

● Déroulement de la séance

- **Pré-test** : Les élèves complètent un pré-test pour recueillir leurs représentations sur la programmation. Ils évaluent leur accord (0 à 5) sur des affirmations telles que "Je me sens capable d'apprendre la programmation informatique" ou "Je pense que la programmation a un intérêt pour mon orientation". Les élèves ayant des connaissances en Python répondent à des questions supplémentaires pour estimer leur niveau.
- **Réalisation des missions Citizen Code Python** :
Les élèves se connectent à Citizen Code Python et accèdent à la rubrique "Mes assignations".



Assignation des missions aux élèves dans l'application Citizen Code Python

- Le gratte-ciel (saison 1, épisode 1) : Initiation à l'interface et découverte des ressources d'aide pour résoudre le problème ; les élèves construisent leur confiance via des séquences d'instructions simples.
 - Le métro (saison 1, épisode 2) : Introduction aux boucles de répétition à travers des situations de débogage.
 - Le totem (saison 1, épisode 3) : Utilisation de boucles pour résoudre un problème plus complexe.
 - La grue (saison 1, épisode 6) : Traduction d'un énoncé en instructions conditionnelles et manipulation de variables.
 - Les enseignes (saison 1, épisode 7) : Mobilisation des notions de conditions, boucles et variables pour un problème combinant plusieurs concepts.
- **Posture d'accompagnement** : Pendant les 15 premières minutes, j'interviens pour aider les élèves dans la prise en main de la plateforme. Après ce temps, mes interventions visent surtout à guider la compréhension des feedbacks et aider au débogage.
 - **Bilan et prolongements individuels** : En fin de séance, les élèves qui ont terminé les cinq missions peuvent tester "La cascade" en Python, introduit par une vidéo explicative de la syntaxe. Ensuite, chaque élève complète un post-test pour voir l'évolution de ses représentations.
 - **Retour en groupe** : La séance se termine par un bilan en classe sur les concepts abordés, la transition collège-lycée, et l'importance de dépasser la programmation par blocs pour adopter un langage textuel.

● Compétences numériques travaillées

Les compétences issues du CRCN travaillées lors de cette séance incluent :

- Information et données : Traiter des données, manipulation de variables et conditions dans un environnement de programmation.
- Communication et collaboration : Partager et publier en ligne, échanger sur des erreurs de programmation et

travailler à leur résolution.

- Création de contenus : Programmer des algorithmes simples via Blockly ou Python.
- Environnement numérique : Résoudre des problèmes techniques, évoluer dans un environnement numérique (connexion via ENT, navigation dans Citizen Code Python).

● Ressources et documents

- [Kit pédagogique](#) Citizen Code Python par Tralalere.
- [Article de présentation de Citizen Code Python](#)
- Synthèse des [TraAMs de l'académie de Poitiers \(2017-2018\)](#) : "Comment assurer la continuité des apprentissages (3ème/2nde) en algorithmique et programmation ?"

● Bilan critique de la séance

Pré-test : les élèves qui ont des notions en Python prennent plus de temps (puisqu'ils ont plus de questions) mais ils sont aussi plus rapides ensuite sur les missions donc cela se compense.

Ces questions supplémentaires me permettront ensuite de mesurer les progrès réalisés dans l'année dans ce langage. Cela m'a permis aussi d'identifier dès à présent les élèves "experts" ou "ressources" sur lesquels je peux m'appuyer, lors de travaux de groupes par exemple.

Engagement des élèves : La plupart des élèves se sont engagés facilement, mais la stratégie d'essai-erreur sans réflexion algorithmique préalable est prédominante. Cela confirme l'importance de travailler explicitement la méthode de résolution algorithmique : phase de recherche, identification des structures et variables, formalisation en langage naturel, traduction en langage de programmation, test, analyse, modifications...

Post-test et feedbacks : En analysant les résultats via la console enseignant, j'ai pu visualiser les erreurs ou "tentatives" de chaque élève. Si la notion de boucle est mieux comprise, celle de condition reste complexe pour certains, comme l'indiquent les nombreuses tentatives observées dans l'activité "La grue".



The screenshot shows the 'SUIVI DES RÉSULTATS' (Tracking Results) section of the Citizen Code teacher interface. It features a search bar with filters for 'SÉLECTIONNER UNE ...' (28) and 'SÉLECTIONNER DES ...' (Tout), and a 'Rechercher' button. Below is a table with columns: Pseudo, Episode, Langage, Activité, Nombre d'heure, and Contexte de réalisation. The table lists six students with their respective assignments and completion status.

Pseudo	Episode	Langage	Activité	Nombre d'heure	Contexte de réalisation
Jérome_18	Saison 1 - épisode 7: Imbrications	Blockly	Les enseignants	8	Dans le cadre d'une assignation
Jasmine_3	Saison 1 - épisode 3: Répéter plusieurs instructions	Blockly	La scène	0	Dans le cadre d'une assignation
Aimée_8	Saison 1 - épisode 3: Répéter plusieurs instructions	Blockly	La scène	0	Dans le cadre d'une assignation
Matthieu_9	Saison 1 - épisode 6: Instructions conditionnelles	Blockly	La grue	2	Dans le cadre d'une assignation
Jérome_18	Saison 1 - épisode 6: Instructions conditionnelles	Blockly	La grue	2	Dans le cadre d'une assignation
Jasmine_3	Saison 1 - épisode 3: Répéter plusieurs instructions	Blockly	L'observatoire	0	Dans le cadre d'une assignation

Interface du professeur dans Citizen Code pour visualiser les résultats des élèves

Les questionnaires pré-test et post-test réalisés auprès de 49 élèves de mes deux classes de Seconde ont révélé une évolution favorable dans presque toutes leurs représentations (voir graphique ci-dessous). Si les garçons ont des représentations envers la programmation globalement plus favorables que les filles au pré-test et au post-test, il est particulièrement prometteur que l'écart entre les filles et les garçons se réduise de près d'un demi point entre les deux tests dans les représentations suivantes :

- *Il est facile d'apprendre la programmation informatique* (-0,4 point d'écart)
- *Je me sens capable d'apprendre la programmation informatique (algorithmique)* (-0,3 point d'écart)
- *J'ai envie de découvrir la programmation informatique (algorithmique)* . (-0,6 point d'écart).

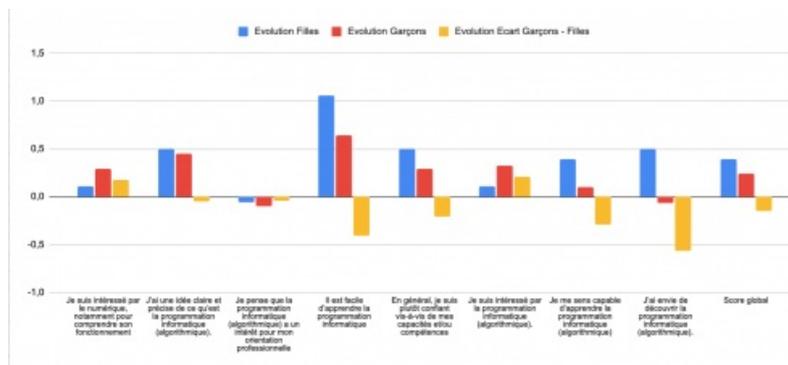


Diagramme de l'évolution des réponses entre le pré-test et le post-test, par genre.

● Conclusion

En guise de première séance dédiée à l'algorithmique et la programmation, l'utilisation de Citizen Code Python a permis d'identifier le niveau et les perceptions des élèves dans ce domaine. L'interface plutôt intuitive et les feedbacks instantanés ont favorisé leur engagement. L'évaluation diagnostique met certes en lumière des degrés divers d'acquisition des notions, souligne une difficulté particulière avec les instructions conditionnelles, mais elle montre aussi que mes élèves savent manipuler des concepts algorithmiques et qu'ils ne partent pas de zéro. Enfin, suite à cette séance, je note une évolution encourageante des perceptions des élèves quant à la programmation, notamment pour les filles (voir peut-être à ce sujet l'article [Inégalités de genre en Mathématiques](#)). Il reste beaucoup à faire pour permettre à mes élèves de maîtriser la résolution algorithmique de problèmes, notamment mathématiques, mais on peut dire que... c'est un bon départ !