

LYCÉE GAY LUSSAC - LIMOGES

- **Présentation de l'équipe du laboratoire**

L'équipe est constituée de 7 Membres , 2 enseignants de Mathématiques de CPGE et 7 enseignants de Mathématique de lycée.

- **Organisation du travail**

Après avoir travaillé une année scolaire (2018-2019) sur le thème de la différenciation, le groupe de travail a souhaité s'orienter sur un autre thème, algorithme et programmation, durant les années scolaires 2019-20 et 2020-21.

Nous avons fait des quatre réunions en présentiel sur l'année 2019-20 et nous avons prévu six réunions en présentiel cette année 2020-21.

Lors de la première réunion annuelle nous fixons des objectifs pour l'année : continuité du travail, les rencontres/formations à définir, les dates des réunions de l'année.

- **Le thème choisi**

Thème : algorithme et programmation au lycée

Objectifs du laboratoire : Dégager des objectifs de réalisation d'algorithmes dans les différents niveaux du lycée et assurer une continuité entre le collège et le lycée et entre le lycée et le supérieur, notamment les classes préparatoires.

Deux intentions : la formation des élèves dans l'esprit des programmes et les épreuves communes de premières technologiques et générales.

- **Lien avec les programmes scolaires / les compétences**

Ce thème s'inscrit dans les programmes, il est clairement explicité dans les programmes de mathématiques du lycée, dans les programmes on trouve des propositions de travaux à organiser dans la classe et dans les documents d'accompagnement des activités pour la classe.

Notre travail du laboratoire s'est donc concentré sur les compétences à acquérir tout au long du lycée, en continuité avec le collège et le post bac, une progression de ces compétences par année scolaire au lycée avec des activités à proposer aux collègues pour faire dans la classe.

Les questions qui se sont posées dans le laboratoire et les réponses associées :

Comment utiliser le langage Python en classe ?

Pour commencer en seconde, on peut utiliser le module Turtle qui permet de revoir les concepts algorithmiques de base (conditions, boucles) vus au collège, il permet aussi de faire un lien avec Scratch, logiciel connu des élèves.

EduPython ou Pyzo semblent adaptés dès la seconde pour mettre en place directement des fonctions qui sont utilisées comme une calculatrice dans le Shell. L'avantage de chacune de ces

distributions est de lire l'interface du code et l'interface du Shell dans la même fenêtre par défaut. EduPython permet de mettre en valeur les erreurs de syntaxes du code.

L'utilisation systématique de fonctions permet de se servir de Python comme de l'utilisation d'une calculatrice où l'élève crée ses propres outils pour répondre à ses besoins. L'élève prend l'habitude d'appeler des fonctions dans le shell.

Un inconvénient du langage Python est que le type de variable n'est pas déclarée, ce qui peut prêter à certaines confusions de types chez les élèves. Cependant, les programmes attendus en mathématiques gèrent principalement des variables de type numérique (entiers ou nombres flottants) et des listes numériques.

L'utilisation de la calculatrice avec Python est un avantage si on n'a pas accès à une salle informatique. Un inconvénient de certaines calculatrice (TI et Casio) est l'édition du code, les commandes sont souvent profondes dans la calculatrice, ça pénalise la rapidité d'écriture d'un programme. Il est préférable de mettre les élèves sur poste pour un gain en efficacité dans la réalisation des programmes.

Utiliser Python sur l'ENT (Jupyter) pour les programmes simples (pas de module Turtle), évite aux élèves d'installer Python chez eux (difficultés pour un grand nombre d'élèves d'installer Python et c'est difficile à gérer par les enseignants). À défaut que les élèves utilisent Python chez eux, il est possible d'utiliser Python sur place au lycée dans les salles d'études ou au CDI.

Il est intéressant de faire travailler les élèves chez eux en programmation pour prolonger ce qui est fait en classe, l'ENT-Jupyter (ou la calculatrice) sont plutôt pertinents pour ce prolongement. D'autre part Jupyter est adapté pour l'utilisation des activités proposées dans les documents d'accompagnement des programmes.

Si possible programmer au fur et à mesure de la progression des autres notions mathématiques, c'est difficile car le programme de mathématiques de seconde est dense. Mais ce principe est intéressant pour faire de la différenciation et appliquer les autres notions mathématiques.

Au lycée Gay-Lussac-Limoges nous disposons d'une salle de 35 postes et 5 salles de 18 postes.

La place du langage naturel et la programmation

Afin de gagner en efficacité dans le traitement, la lecture d'un algorithme, il est souhaitable de ne pas séparer le langage naturel du programme en Python. On gagne ainsi un temps considérable sur la mise en place d'algorithmes et sur ce qui est attendu dans les programmes.

Pour la différenciation et une meilleure compréhension des deux attentes on peut mettre en face l'algorithme en langage naturel et le programme attendu, y compris lors des évaluations. L'élève prend l'habitude des deux lectures, il peut rapidement faire le lien entre chacune des écritures.

On ne peut pas se soustraire du langage naturel qui apporte une bonne compréhension d'un algorithme à l'écrit, c'est un axe de réflexion pour construire un programme et il est un objectif pour la formation notamment pour viser des attendus dans les sujets du baccalauréat en épreuve commune.

Le langage naturel peut être vite contraignant si on doit s'emparer de plusieurs fonctions, il convient de séparer les objectifs algorithmiques dans un "grand programme".

Formation

Suite à la lecture des documents d'accompagnement de seconde, Benoît Rivet a proposé un exposé sur les erreurs numériques en Python, les difficultés pour interpréter des nombres flottants non attendus. La méthode de balayage pour trouver les décimales de racine de 2.

Le but n'est pas de présenter aux élèves le développement des erreurs d'approximation en Python, mais que chaque enseignant soit pleinement conscient des limites dans l'utilisation de Python pour par exemple approcher des valeurs comme racine de 2. (en annexe pour le développement).

Les objectifs par niveau :

Dans les programmes les attentes des savoir-faire des élèves ne sont pas clairement explicitées exemple en première générale spécialité ;

Algorithmique et programmation

La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. La classe de seconde a permis de consolider les acquis du cycle 4 autour de deux idées essentielles :

- la notion de fonction ;
- la programmation comme production d'un texte dans un langage informatique.

L'enseignement de spécialité de mathématiques de classe de première vise la consolidation des notions de variable, d'instruction conditionnelle et de boucle ainsi que l'utilisation des fonctions. La seule notion nouvelle est celle de liste qui trouve naturellement sa place dans de nombreuses parties du programme et aide à la compréhension de notions mathématiques telles que les suites numériques, les tableaux de valeurs, les séries statistiques...

Comme en classe de seconde, les algorithmes peuvent être écrits en langage naturel ou utiliser le langage Python.

Les notions relatives aux types de variables et à l'affectation sont consolidées. Comme en classe de seconde, on utilise le symbole « ← » pour désigner l'affectation dans un algorithme écrit en langage naturel.

L'accent est mis sur la programmation modulaire qui permet de découper une tâche complexe en tâches plus simples.

On peut lire des attendus sur les contenus.

Nous avons souhaité uniformiser nos attentes en termes de production d'élèves :

Outre les attentes dans les programmes, on peut distinguer des évolutions de codages suivants le niveau des élèves.

- En seconde l'objectif principal est la bonne lecture d'un algorithme. Il est cependant important de comprendre aussi la construction d'un algorithme et d'un programme.
- En première spécialité mathématiques les élèves devraient être capable de compléter et modifier un programme.
- En première technologique les élèves devraient être capable de compléter et modifier un programme.
- En terminale spécialité mathématiques commencer à faire des programmes seuls, sur des algorithmes de bases (comme un algorithme de seuil) et savoir écrire l'algorithme associé.
- En terminale technologique s élèves devraient être capable de compléter et modifier un programme
- En CPGE avoir faire un programme, savoir analyser un programme.

Des activités ont été développées pour répondre aux objectifs de la programmation par niveau, ces activités sont disponibles en annexe.

- **activités**

Les activités :

- Activités mentales
- Exercices (souvent ceux du livre)
- Devoirs
- Travaux dirigés
- Découper un problème complexe en tâches simples : décrire les fonctions à utiliser pour répondre à la question ?

Annexes des activités - commentaires :

- Exemple 1 en seconde – boucle :
 - Utilisation du module Turtle
 - Découpage d'un problème complexe en un tâche simple en faisant appel aux fonctions réalisées. On peut réaliser le programme complet de l'exercice 1 en faisant des boucles de boucles ou en appelant des fonctions, ici l'approche favorise l'appel de fonctions.
 - Écriture de l'algorithme et lecture de programmes, programmes à compléter (l'objectif de compléter le programme est de comprendre le statu des variables, ce qui améliore la lecture d'un programme).
 - Rendre plus autonome l'élève en se servant du travail accompli pour se l'approprier et faire un programme et un algorithme.
 - Bilan de l'activité : fonctionne plutôt bien, la prise d'autonomie et l'appropriation des commandes n'est pas évidente pour les élèves.

- Exemple 2 en seconde :
 - Prise d'initiatives, permet la différenciation par les contenus
 - Faire des mathématiques : calcul de longueur (géométrie de Pythagore) ou travailler sur les coordonnées des milieux.
 - Bilan de l'activité : trop difficile à mettre en place, mais le recherche en groupe permet de dégager les deux algorithmes aux approches géométriques différentes. La recherche collaborative a été intéressante à mener.

- Exemple 1 en première générale - le flocon de Von-Koch :
 - Exercice classique qui permet de travailler les suites numériques notamment géométriques,
 - Permet de faire de la différenciation sur les approches,
 - Découpages de tâches d'un problème complexe en complétant les affectations des variables au fur et à mesure du problème, identifications des variables en jeux
 - Il est possible d'utiliser le tableur pour faire ce problème, le glisser-coller représente alors une boucle qui incrémente les variables en jeux.
 - Bilan de l'activité :

- Exemple 2 en première générale - approximation de Pi :

- Utilisation de GeoGebra pour mettre en place un algorithme d'approximation (différenciation, repérage des variables, mise en place de boucle)
 - Lecture d'algorithmes et utilisation
 - Le programme Python devient un prolongement d'un travail sur GeoGebra
 - Bilan de l'activité : neant (jamais testé, la situation sanitaire ne l'a pas permis)
- Exemple 1 en terminale technologique STMG :
 - Suite et tableur, la mise en place sur tableur permet de comprendre les variables de l'algorithme, les boucles de l'algorithme. (différenciation des processus)
 - Travailler les suites numériques autrement et donner du sens aux relations, notamment la relation de récurrence d'une suite arithmétique.
 - La fonction print permet de retrouver d'afficher les résultats intermédiaires et de retrouver les valeurs du tableur.
 - Bilan de l'activité : fonctionne bien permet de mettre les élèves en confiance, le tableur est un outil plutôt bien maîtrisé par les élèves.
- Exemple 2 en terminale technologique STMG :
 - Répétition du modèle de travail précédent, permet d'attendre plus d'autonomie de la part des élèves notamment pour compléter les algorithmes.
 - Suite et tableur, la mise en place sur tableur permet de comprendre les variables de l'algorithme, les boucles de l'algorithme. (différenciation des processus)
 - Travailler les suites numériques autrement et donner du sens aux relations, notamment la relation de récurrence d'une suite géométrique.
 - La fonction print permet de retrouver d'afficher les résultats intermédiaires et de retrouver les valeurs du tableur.
 - Bilan de l'activité : fonctionne bien permet de mettre les élèves en confiance, le tableur est un outil plutôt bien maîtrisé par les élèves.
- Exemple 3 en terminale technologique STMG :
 - Répétition du modèle de travail précédent, permet d'attendre plus d'autonomie de la part des élèves notamment pour compléter les algorithmes.
 - Suite et tableur, la mise en place sur tableur permet de comprendre les variables de l'algorithme, les boucles de l'algorithme. (différenciation des processus)
 - Travailler les suites numériques autrement et donner du sens aux relations, notamment la relation de récurrence d'une suite géométrique.
 - La fonction print permet de retrouver d'afficher les résultats intermédiaires et de retrouver les valeurs du tableur.
 - Bilan de l'activité : fonctionne bien permet de mettre les élèves en confiance, le tableur est un outil plutôt bien maîtrisé par les élèves.
- Exemples de questions d'activités mentales en première :

Algorithme :

$u \leftarrow 10$

$u \leftarrow u + 2$

$u \leftarrow u + 4$

○

Que vaut u à la fin de l'algorithme ?

Algorithme :

$n \leftarrow 0$

$u \leftarrow 10$

$n \leftarrow n + 1$

$u \leftarrow u + n$

○

Que vaut u à la fin de l'algorithme ?

Algorithme :

$n \leftarrow 0$

$u \leftarrow 10$

Tant que $u < 29$ faire

$n \leftarrow n + 1$

$u \leftarrow u + 5$

Fin Tant que

Que valent n et u à la fin de l'algorithme ?

○

Algorithme :

$u \leftarrow 10$

Pour n variant de 1 à 3 :

$u \leftarrow u + 4$

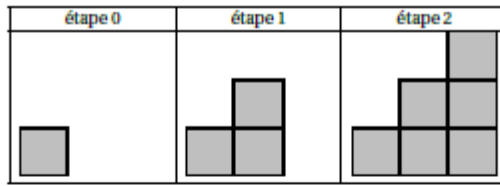
Fin pour

Que vaut u à la fin de l'algorithme ?

○

- Exemple d'exercice en devoir en première :

On considère la construction suivante :



On note u_n le nombre de carrés qui forment un motif à l'étape n . Ainsi $u_0 = 1$, $u_1 = 3$ et $u_2 = 6$.

1. En poursuivant le principe de construction, quel serait le nombre de carrés qui composeraient le motif de l'étape 4 ?
2. Exprimer u_{n+1} en fonction de u_n et de n .
3. On donne l'algorithme suivant et le tableau qui permet de suivre l'exécution de l'algorithme :

```

n ← 0
u ← 1
Tant que u < 30 faire
  n ← n + 1
  u ← u + n + 1
Fin tant que
    
```

n	u	condition $u < 30$
0	1	vraie
1	3	vraie
2	6	vraie
...

Recopier et compléter le tableau jusqu'à déterminer les dernières valeurs de u et de n calculées par l'algorithme.

- **La suite du travail**

Objectifs 2020-21 :

- Poursuivre le travail commencé l'année 2019-20, notamment en alimentant les ressources des classes de terminales
- Objectifs de formations :
 - Liaison lycée - classe préparatoire : discuter des "besoins" du supérieur, établir des objectifs de formation au lycée.
 - Liaison collège-Lycée : inviter des collègues de collèges du bassin de Gay-Lussac pour échanger, proposer des idées pour la formation des élèves en algorithmique et pourquoi pas créer une activité de fin de troisième et une activité de début de seconde.