



Activités en classe pour travailler la programmation en Scratch et Python du collège au lycée

Février 2025

Recueil d'expérimentations menées par l'équipe de professeurs du bassin Antibes-Valbonne-Vallauris ayant participé à la formation du laboratoire de mathématiques du lycée Simone Veil pendant l'année 2023-2024 en collaboration avec le Groupe Numérique Disciplinaire de mathématiques de l'académie de Nice

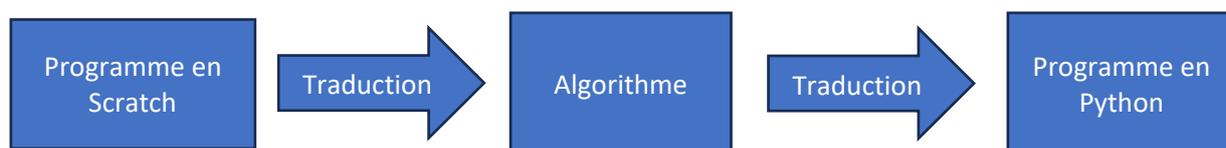
PREAMBULE

Le choix du thème de formation fait par les stagiaires sur la programmation en Scratch et Python au collège et au lycée n'est pas anodin. La difficile transition entre Scratch et Python lors du passage du collège au lycée vécu en classe de seconde par les élèves méritait d'y consacrer une formation.

Cette brochure est l'aboutissement du travail réalisé pendant cette formation. Il prend la forme d'un recueil d'activités toutes conçues et expérimentées dans leurs propres classes par les professeurs qui y ont participé.

Les enseignements tirés de la première demi-journée de formation avaient mis en avant un certain nombre de points nécessaires à une transition efficace dans l'apprentissage de la programmation dans le cours de mathématiques entre le collège et le lycée, pour faciliter le transfert de Scratch à Python pour les élèves :

- Un travail régulier en programmation Scratch tout au long de la scolarité au collège
- Un travail de liaison collège-lycée axé sur le travail entre pairs
- Des activités de transition entre Scratch et Python utilisant une traduction « algorithmique » suivant le schéma suivant :



- Des activités contextualisées pour faciliter la compréhension des élèves et susciter leur intérêt notamment chez les filles.

Il sera aussi question dans ce recueil de présenter différentes modalités présentées aux élèves pour réaliser ces activités en classe en proposant des alternatives à la salle informatique. Il ressort souvent que l'aspect matériel est parfois un frein pour les professeurs pour mettre en place ces activités. Celles-ci ont donc été pensées pour être plus efficaces et surtout plus faciles à installer au sein de la classe aussi bien pour le professeur que pour les élèves.

Professeurs Formateurs

Sylvain ETIENNE, professeur de mathématiques au collège L'Eganaude à Biot (Alpes-Maritimes) et formateur académique, membre de l'équipe du laboratoire de mathématiques du lycée Simone Veil.

Sandrine PIC, professeure de mathématiques au Collège International de Valbonne (Alpes Maritimes) et formatrice pour le Groupe Numérique Disciplinaire (G.N.D.) de mathématiques.

Olivier PILORGET, professeur de mathématiques au collège Sydney Bechet à Antibes (Alpes Maritimes), Interlocuteur Académique au Numérique, formateur pour le pour Groupe Numérique Disciplinaire (G.N.D.) de mathématiques.

Professeurs de mathématiques auteurs d'articles :

Edith BONNERY (Collège Sidney Bechet, Antibes) ; Denis RIGUCCINI (Lycée Audiberti, Antibes) ; Françoise HELSON (Lycée S. Veil, Valbonne) ; Saloua RHIOUI (Lycée S. Veil, Valbonne) ; Emma GILESTRI (Lycée S. Veil, Valbonne) ; Frank D'ESQUERMES (Lycée S. Veil, Valbonne) ; Isabelle MOURARD (Lycée S. Veil Valbonne).

Inspecteurs d'Académie - Inspecteurs Pédagogiques Régionaux

Clarisse FIOL et Cédric GOURJON, IA-IPR de mathématiques, pilotes du Groupe Numérique Disciplinaire (G.N.D.) de mathématiques de l'académie de Nice.

TABLE DES MATIERES

PREAMBULE	2
TABLE DES MATIERES	3
I – ACTIVITES DE PROGRAMMATION EN SCRATCH AU COLLEGE	4
ACTIVITE 1: ACTIVITE SCRATCH SUR LES PRIORITES OPERATOIRES	5
ACTIVITE 2: ACTIVITE SCRATCH SUR LES PROGRAMMES DE CALCUL	8
ACTIVITE 3: ACTIVITE SCRATCH SUR LES TRACES DANS UN REPERE AVEC LE LUTIN	14
II – ACTIVITES DE TRANSITION DE SCRATCH A PYTHON	20
ACTIVITE 4: PROGRAMMATION SCRATCH AU COLLEGE/LYCEE D’UNE APPLICATION « SANTE »	21
ACTIVITE 5: PROGRAMMATION SCRATCH AU COLLEGE/LYCEE A TRAVERS LA CONSTRUCTION DE FIGURES GEOMETRIQUES	25
ACTIVITE 6: INTRODUCTION DU LANGAGE PYTHON EN CLASSE DE SECONDE	29
III – ACTIVITES DE PROGRAMMATION EN PYTHON AU LYCEE	33
ACTIVITE 7: ACTIVITE SUR LA DETERMINATION D’UN EXTREMUM EN CLASSE DE SECONDE	34
ACTIVITE 8 : VARIATIONS ET NOTION DE LIMITES D’UNE SUITE, NIVEAU PREMIERE	39
ACTIVITE 9: UTILISATION DES NOTEBOOKS CAPYTALE POUR 3 ACTIVITES MELANT HISTOIRE DES MATHEMATIQUES ET PROGRAMMATION PYTHON EN TERMINALE	45

I – ACTIVITES DE PROGRAMMATION EN SCRATCH AU COLLEGE

ACTIVITE 1: ACTIVITE SCRATCH SUR LES PRIORITES OPERATOIRES

Nom / Etablissement : Sandrine PIC – Collège International de Valbonne

Niveau : 5^e

Type d'activités/modalités de travail : Travail en demi-groupe en salle informatique.

Lien vers l'activité :

https://mathsavesnes.etab.ac-lille.fr/pdf/5eme/tn2_activite_scratch_priorites_de_calcul.pdf

<https://college-montaigne.fr/wp-content/uploads/3%C2%B0-AP-priorit%C3%A9s-op%C3%A9rateurs-et-scratch-.pdf>

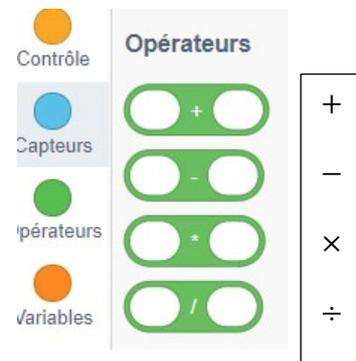
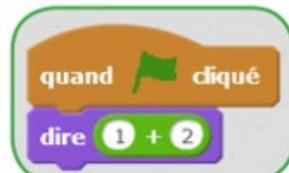
Thématique(s) du programme – Objectifs :

- Mathématiques :
 - Priorités opératoires
- Algorithmique / Informatique :
 - Utilisation des blocs pour construire des opérations – Notion de priorité (calcul le plus à l'intérieur).
 - Entrées/sorties : afficher le résultat, formater l'affichage.

CALCULER AVEC SCRATCH

Les opérations sont accessibles dans le menu Opérateurs.

Pour afficher le résultat d'un calcul, il faut avec un lutin, utiliser le bloc « Dire »

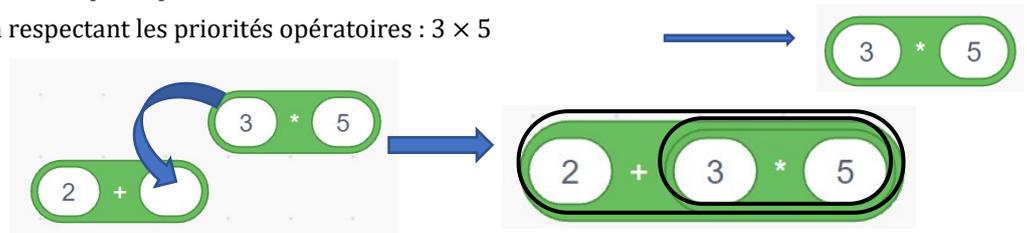


ORGANISER UN CALCUL

On souhaite programmer le lutin pour qu'il effectue le calcul suivant : $2 + 3 \times 5$

On programme le lutin en respectant les priorités opératoires : 3×5

Puis : $2 + 3 \times 5$



A toi de remplir le tableau :

Blocs	Expressions	Résultat

PROGRAMMER UN CALCUL

Représente les blocs correspondants à l'expression, puis teste avec Scratch

Blocs	Expression	Résultat
	$1 + 10 \div 2$	
	$(1 + 2) \times (8 - 6)$	
	$1 + 2 \times 3 - 4$	
	$2 \times [(3 + (4 - 9 \div 3)) - 2]$	

ANNONCER UN RESULTAT AVEC UNE PHRASE REPONSE

Pour associer une phrase réponse à un résultat, on utilise le bloc « Regrouper », dans le menu « Opérateurs ».

Exemple :



Remarques :

- La première partie du bloc sert à énoncer la phrase, et la seconde à insérer le calcul.
- On peut insérer un bloc regrouper dans un bloc regrouper.

RESOLUTION DE PROBLEMES

A toi de résoudre le problème suivant avec Scratch :

« J'ai acheté 5 bouteilles de soda à 0,65 € l'unité, et 4 paquets de sucreries à 1,25 € l'unité.

Combien ai-je payé au total ? »

1/ Quelle est l'expression numérique ? $P =$

2/ La phrase réponse est : « Je vais payer en euros : ... »

Représente le bloc regrouper que tu vas associer au lutin qui dit la phrase réponse et le résultat.

3/ On souhaite maintenant faire dire au lutin : « Je vais payer... € » ? C'est-à-dire que l'on veut insérer le calcul avant le symbole €. Pour cela, on va imbriquer un bloc regrouper dans un bloc regrouper. Programme-le et représente les blocs utilisés :

ANALYSE/PRODUCTION

L'activité se déroule en trois parties.

- Une première débranchée où le lien est fait entre écriture et bloc.

ORGANISER UN CALCUL

- La seconde permet de traduire des expressions avec des blocs et de programmer afin de vérifier le calcul.

PROGRAMMER UN CALCUL

- Quelques difficultés dans cette partie, sur l'ordre d'imbrication des blocs. Les élèves ont du mal à vérifier le résultat du programme avec la calculatrice ou bien mentalement. Ils prennent pour acquis le résultat.
- Enfin une dernière partie dédiée à la résolution de problème simple, avec Scratch afin d'utiliser les blocs d'entrées/sorties.
Pas de difficultés supplémentaires, ils comprennent assez rapidement qu'il est nécessaire d'imbriquer les blocs « regrouper » afin de produire une phrase plus complexe.

BILAN

a) Objectifs atteints

A l'issue de la séance, les élèves ont compris comment gérer les priorités opératoires avec Scratch. La notion du « calcul le plus à l'intérieur » est vue avec les blocs et les parenthèses. Ils savent restituer un résultat par le biais du lutin, en faisant une phrase.

b) Limites

Il n'y a pas d'apprentissage majeur sur la partie algorithmique. Les élèves poursuivent la découverte de Scratch

c) La suite

Avec le calcul littéral, poursuivre le travail sur les Entrées/Sorties (bloc demander et attendre, variable réponse).

ACTIVITE 2: ECRIRE DES PROGRAMMES DE CALCUL AVEC SCRATCH

Objectifs pédagogiques : L'objectif de cette activité est de créer des programmes informatiques avec le logiciel Scratch permettant d'afficher des résultats de programmes de calcul.

Nom / Etablissement : Olivier Pilorget - Collège Sidney Bechet

Niveau : 5^e

Type d'activités/modalités de travail : L'activité nécessite que l'élève dispose d'une tablette et d'un accès à internet. L'énoncé est distribué au format papier et contient un certain nombre de liens vers l'interface de programmation CODABLOC (similaire à SCRATCH) accessible via CAPYTALE sur l'ENT. Les fichiers des élèves peuvent ainsi être récupérés pour être évalués par la suite.

Thématique(s) du programme – Objectifs :

Cette activité propose d'élaborer et de programmer des « programmes de calcul ». Les thèmes mathématiques abordés dans l'activité sont écrire, mettre au point et exécuter un programme.

Capacités Mathématiques :

Appliquer un programme de calcul. Construire et s'appropriier la notion de variable.

Capacités Algorithmique / Informatique :

Blocs d'entrées et de sorties, notion de variable, utiliser les blocs opérateurs pour élaborer un programme de calcul.

PRESENTATION DE L'ACTIVITE

1ere Partie : Travailler sur la notion de variable

Cette première partie de l'activité se déroule en interaction avec le groupe classe.

L'objectif est de réfléchir, en **activité débranchée**, sur la notion de variable : un même script permet d'effectuer les calculs pour n'importe quel nombre de départ.

Au niveau de la programmation, le statut particulier des blocs « résultats » et « réponse » ainsi que les blocs opérateurs sont abordés. L'utilisation du bloc « regrouper » permet l'interaction avec l'utilisateur pour annoncer le résultat avec une phrase réponse.



2e Partie : Compléter un script

L'élève, en autonomie, réinvestit le travail réalisé à la question 1 en se concentrant uniquement sur les notions de variables et de bloc opérateur. L'idée est de compléter un script pour qu'il corresponde au programme de calcul proposé. Un test permet de vérifier le bon fonctionnement du script.





3e Partie : Remettre les différents blocs d'un script dans l'ordre

Un programme de calcul plus complexe est proposé. On demande à l'élève d'organiser un script dont on a déjà tous les blocs à disposition (pas de perte de temps à rechercher les blocs). L'élève va devoir faire attention à bien positionner les blocs « réponse » et « résultat ».

Un test à effectuer avec un nombre est demandé pour vérifier que le script correspond bien au programme de calcul.

Une **trace écrite** est alors élaborée avec le groupe classe :

Proposition de trace écrite :

Faire le point sur la notion de variable

Pour écrire le script avec Scratch on a **créé la variable** **résultat** à partir du menu Données. Une variable est une « boîte » dans laquelle on stocke une information (ici un nombre) pour l'utiliser plus tard. On désigne une variable par un nom.

! Le bloc **réponse** se trouve dans le menu « Capteurs ». Quand l'utilisateur saisit la réponse à la question « quel est le nombre de départ », cette dernière est automatiquement mise dans la variable « réponse ». On peut alors réutiliser cette réponse en utilisant ce bloc.

4e Partie : Ecrire un script qui permet de proposer un programme de calcul à un utilisateur.

A ce stage, l'élève dispose d'un programme de calcul à respecter et un « scénario utilisateur » est imposé. Il lui est demandé de réinvestir tout ce qui a été vu dans les questions précédentes pour écrire le script dans son intégralité.

Un prolongement en devoir hors classe est proposé à la fin de l'activité.

***Point de vigilance concernant les « liens CAPYTALE »
présents dans l'activité***

Vous trouverez ci-dessous les liens « publics » pour chaque question. Ils permettent d'accéder aux différents scripts, de les sauvegarder pour pouvoir les partager avec les élèves par la suite.

Question 1 : <https://capytale2.ac-paris.fr/web/b/3678396>

Question 2 : <https://capytale2.ac-paris.fr/web/b/3622414>

Question 3 : <https://capytale2.ac-paris.fr/web/b/3622604>

Question 4 : <https://capytale2.ac-paris.fr/web/b/3622941>

Question 4 dhc : <https://capytale2.ac-paris.fr/web/b/3659642>

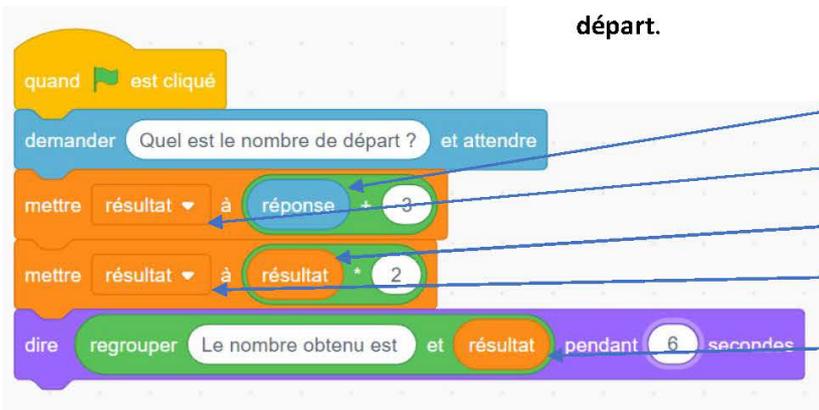


PROGRAMMER DES PROGRAMMES DE CALCUL

L'objectif de cette activité est de créer des programmes informatiques avec le logiciel Scratch permettant d'afficher des résultats de programmes de calcul.

1. Voici un programme de calcul et le script correspondant sur Scratch :

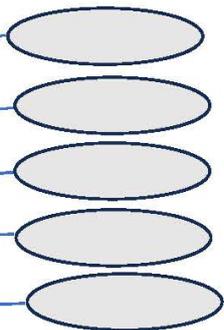
- Choisir un nombre de départ
- Ajouter 3 à ce nombre
- Multiplier le résultat par 2
- Indiquer le nombre obtenu



a. Quel nombre obtient-on si on choisit 13 au départ ?

.....

b. Indiquer dans chaque bulle la valeur des deux variables tout au long du script lorsque l'on choisit 9 comme nombre de départ.



Dans la suite de l'activité, vous effectuerez le travail sur l'application Capytale (Accessible via l'ENT).

Vous pouvez vous connecter à l'activité en utilisant le lien internet, le code de l'activité ou le QR code associé. Ne pas oublier de RENDRE votre copie numérique.

c. Le script ci-dessus est accessible (lien ci-contre) avec l'application Capytale.

Consigne :

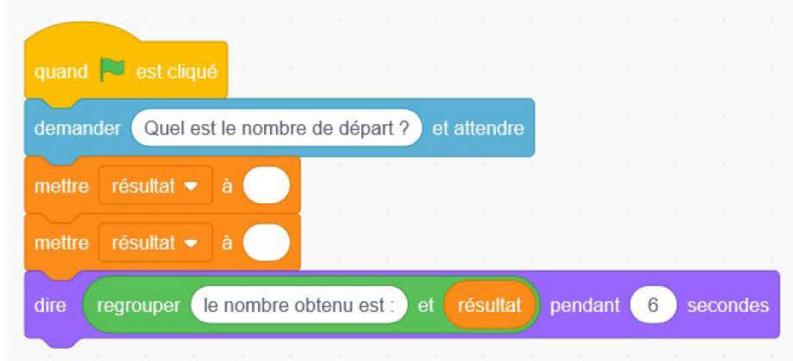
A l'aide de ce programme (prog1), compléter le tableau ci-dessous :

Nombre de départ	7	12	20	33
Résultat du programme de calcul				

Lien sur Capytale : [Prog1](#)
code : 2b74-3622414

2. Compléter un programme de calcul

- Choisir un nombre de départ
- Prendre le double de ce nombre
- Soustraire 5
- Indiquer le nombre obtenu



Consigne :

a. Sur l'application Capytale, **compléter le programme Scratch**, en utilisant des opérateurs, pour qu'il corresponde au programme de calcul ci-dessus.

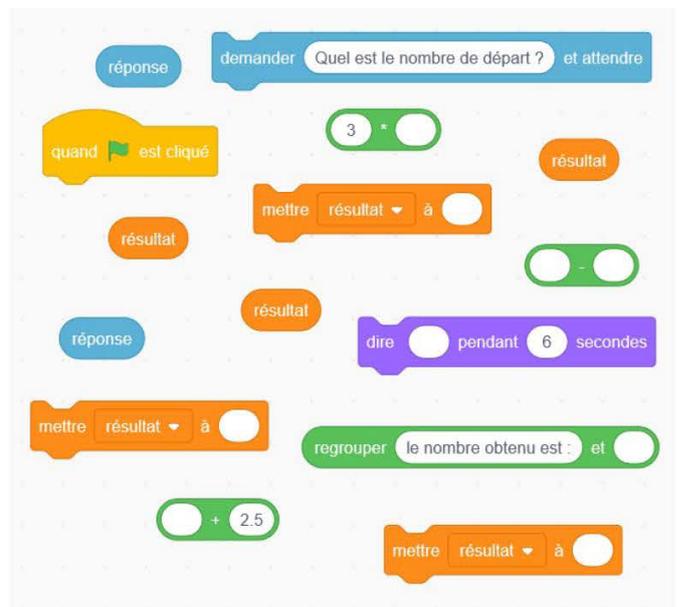
b. Tester le programme de calcul avec le nombre 25.
Quel est le résultat donné par le programme ?

Lien sur Capytale : [Proq2](#)
code : 2b74-3622414



3. Programme de calcul : remettre les différents blocs dans l'ordre

- Choisir un nombre de départ
- Prendre le triple de ce nombre
- Ajouter 2,5
- Soustraire le nombre de départ
- Indiquer le nombre obtenu



Consigne :

a. Sur l'application Capytale, **remettre les blocs dans le bon ordre** pour que le script corresponde au programme de calcul.

b. En utilisant votre script sur Scratch, tester le programme de calcul avec le nombre 3.
Quel est le résultat donné par le programme ?

Lien sur Capytale : [Proq3](#)
code : b556-3622604



4. Elaborer un programme de calcul et le tester

a. Ecrire un script qui permet de proposer le programme de calcul ci-dessous à un utilisateur.

- Choisir un nombre de départ
- Multiplier ce nombre par 4
- Ajouter 8
- Soustraire le nombre de départ
- Indiquer le nombre obtenu



Lien sur Capytale : [Proq4](#)
code : c53d-3622941



On souhaite avoir le scénario suivant :

- Créer la variable résultat dans l'onglet « Variables »
- Le programme commence quand le drapeau vert est pressé.
- **Le lutin dit** pendant 5 secondes : « *Je vais te proposer un programme de calcul* »
- **Il demande** « *choisis un nombre* » et attend.
- **Il dit** « *Multiplie ce nombre par 4* » pendant 5 secondes
- **Il dit** « *Ajoute 8 à ce nombre* » pendant 5 secondes
- **Il dit** « *Pour terminer soustrais à ce résultat le nombre de départ* » pendant 5 secondes
- **Il dit** alors « *Le résultat final est* » pendant 5 secondes
- **Le lutin affiche** alors la bonne réponse.

b. Tester le programme de calcul avec le nombre 20.
Quel est le résultat donné par le programme ?

Prolongement en devoir hors classe :

Modifier le script précédent pour qu'il corresponde au même programme de calcul mais qu'à la fin, plutôt que de proposer la réponse, le lutin demande à l'utilisateur d'écrire son résultat. Le lutin devra alors vérifier si la réponse de l'utilisateur est correcte ou non.

Scénario possible :

- Le lutin demande « *Quel est le résultat de ce programme de calcul ?* » et attend.
- Si la réponse de l'utilisateur est la bonne il dit « *Bravo tu es un champion !* » pendant 5 secondes.
- Sinon il dit « *Et non, tu t'es trompé(e) !* » pendant 5 secondes. Puis il ajoute « *La bonne réponse est :* » et le lutin affiche alors la bonne réponse.

Lien sur Capytale : [Proq hors classe](#)
code : 2b74-3622414



ACTIVITE 3: ACTIVITE SCRATCH SUR LES TRACES DANS UN REPERE AVEC LE LUTIN

Nom / Etablissement : Sandrine PIC – Collège International de Valbonne

Niveau : 3^e

Type d'activités/modalités de travail : Travail en demi-groupe en salle informatique.

Lien vers l'activité : <https://www.monclasseurdemaths.fr/profs/algorithmique-scratch/>

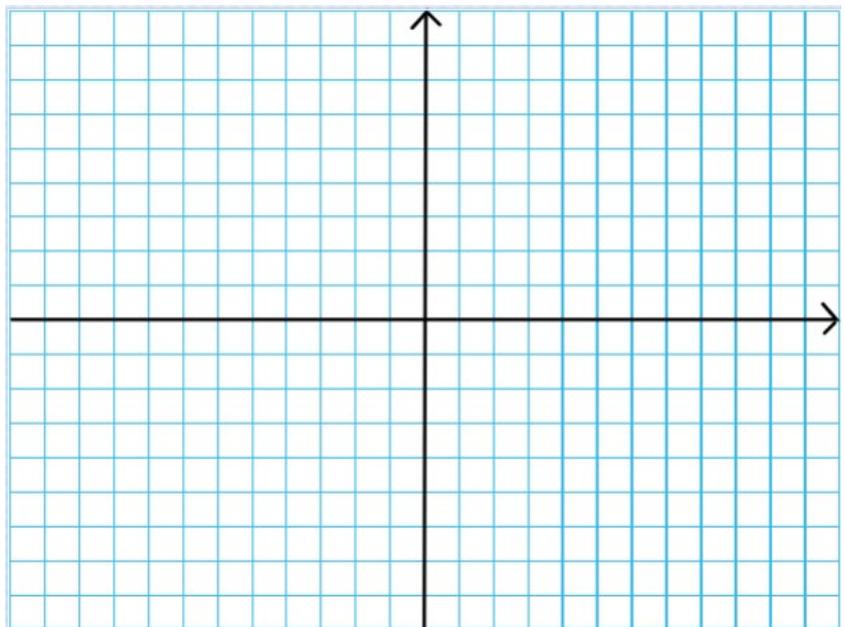
Thématique(s) du programme – Objectifs :

- Mathématiques :
- Repérage
- Algorithmique / Informatique :
- utilisation des blocs de déplacement
- utilisation du stylo
- Itérations l'algorithme.

PREMIERE PARTIE – Comprendre un script

Pour chacun des cas :

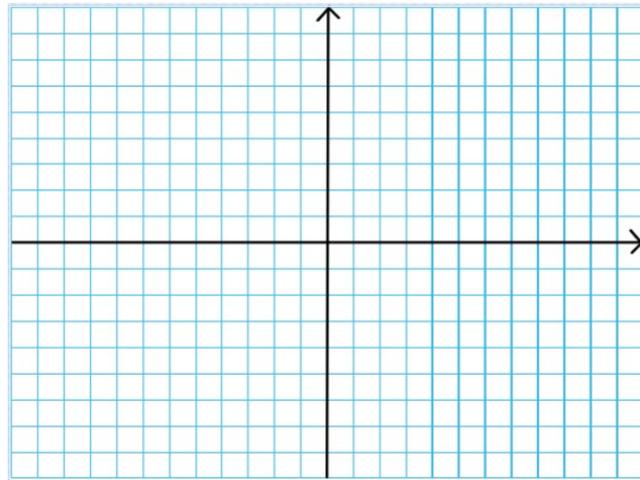
- 1) Dessine sur la grille la figure que va tracer le lutin en respectant les positions et les dimensions.
Unité : 1 carreau pour 20 pas en abscisse et en ordonnée.
- 2) Vérifie ton tracé en programmant le script.
Importe le fichier arrière-plan dans ton dossier depuis Pearltrees : « 3^{ième} /Scratch/3 Figure Fond »
Puis importe-le dans ton projet.



```

quand est cliqué
  aller à x: -140 y: 0
  s'orienter en direction de 90
  effacer tout
  stylo en position d'écriture
  répéter 5 fois
    répéter 4 fois
      avancer de 40 pas
      tourner de 90 degrés
    avancer de 60 pas
  stop tout

```



```

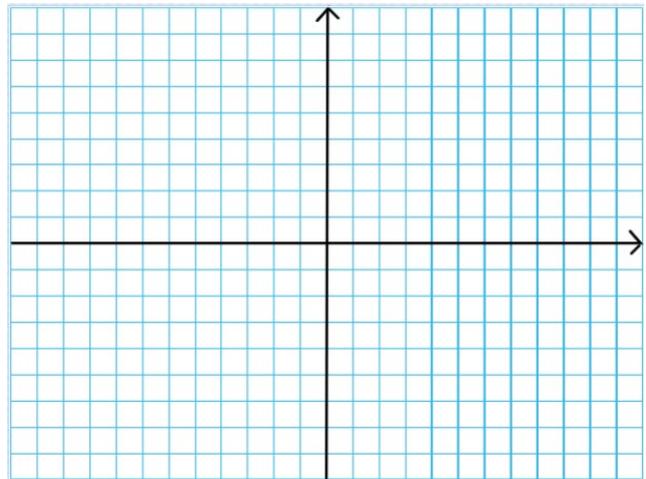
quand est cliqué
  aller à x: -80 y: 80
  s'orienter en direction de 90
  effacer tout
  stylo en position d'écriture
  répéter 4 fois
    Petit carré
    avancer de 160 pas
    tourner de 90 degrés
  stop tout

```

```

définir Petit carré
  répéter 4 fois
    avancer de 60 pas
    tourner de 90 degrés

```



```

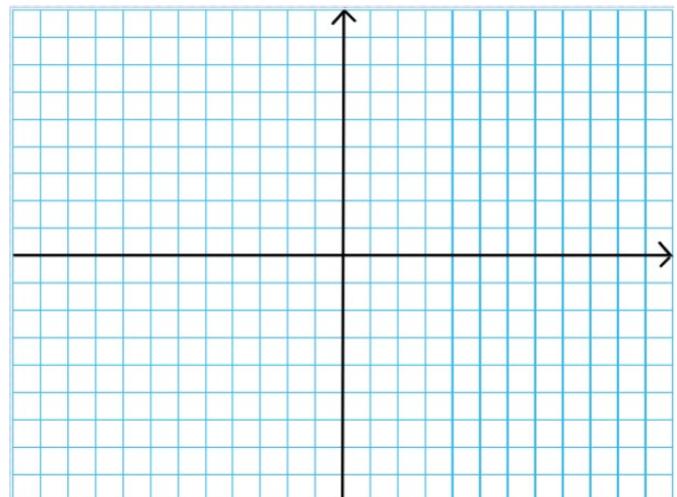
quand est cliqué
  Initialisation
  répéter 4 fois
    avancer de Longueur pas
    tourner de 90 degrés
    ajouter 20 à Longueur
  stop tout

```

```

définir Initialisation
  aller à x: -200 y: 0
  s'orienter en direction de 90
  mettre Longueur à 20
  effacer tout
  stylo en position d'écriture
  stop ce script

```



ANALYSE/PRODUCTION

L'activité a été proposée en demi-groupe, en salle informatique, pour deux classes de 3^{ème}. Les élèves ont tous fait la première partie. Pour la seconde, ils ont presque tous fait le premier script, certains ont fini le second et deux élèves sont venus presque à bout du 3^{ème}.

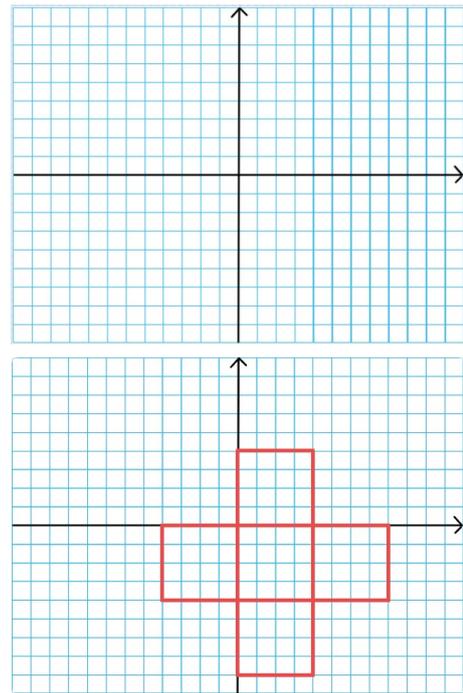
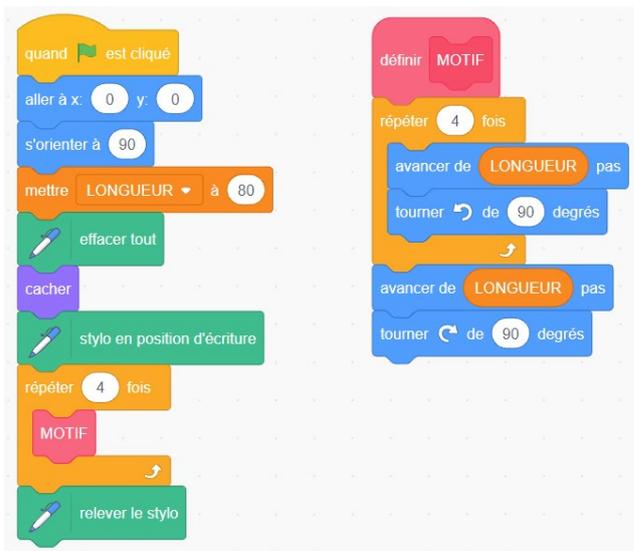
- La première partie : certains ont eu besoin de se lever, de mimer le déplacement. Quelques confusions entre droite et gauche.
- Seconde partie : le test n'est pas automatique. Les élèves codent tout, testent et ont beaucoup de mal à comprendre leurs erreurs. Ils ne séparent pas les problèmes, par exemple en supprimant une itération, en simplifiant pour tester des bouts de scripts. Cette partie est laborieuse.

Un exercice similaire a été proposé dans une évaluation aux deux classes.

Classe 1 : énoncé

Dessine sur la grille la figure que va tracer le lutin en respectant les positions et les dimensions.

Unité : 1 carreau pour 20 pas en abscisse et en ordonnée.



Productions et Bilan

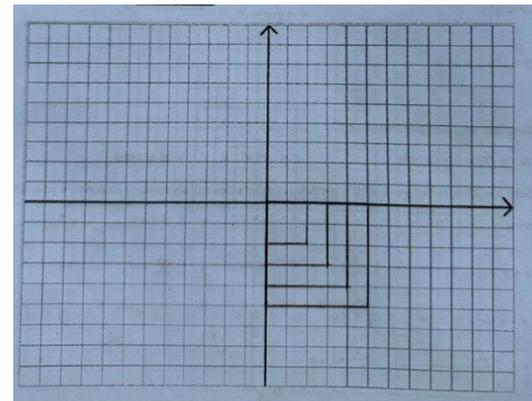
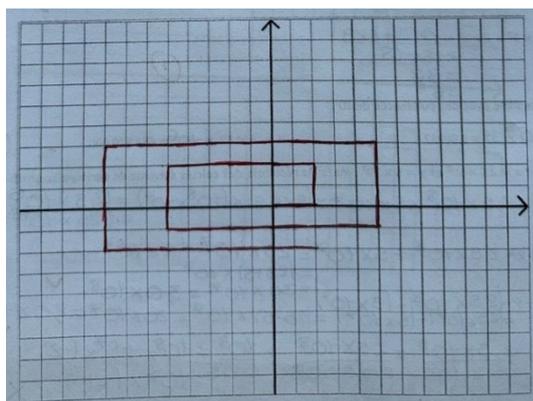
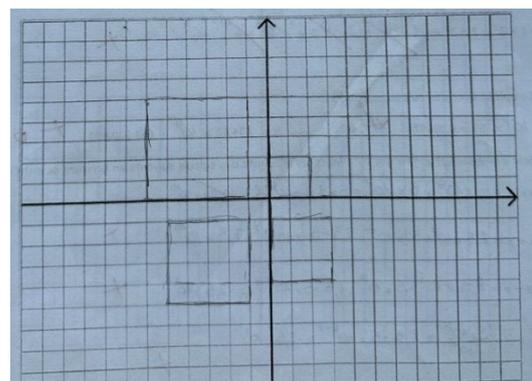
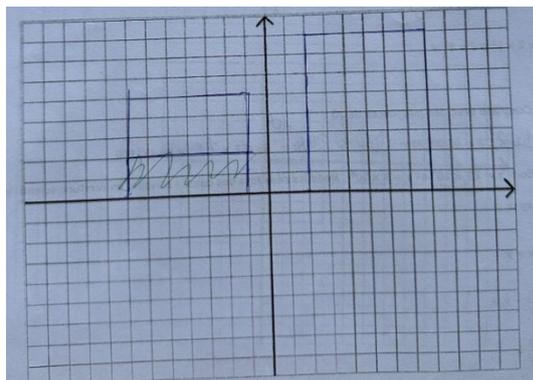
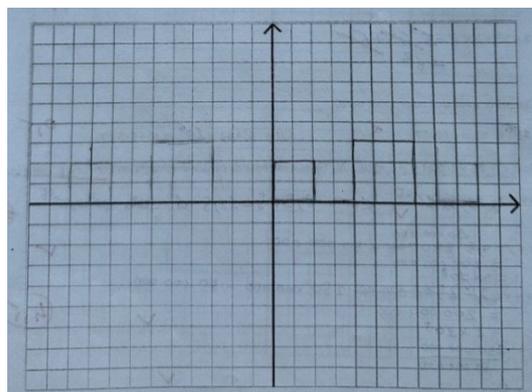
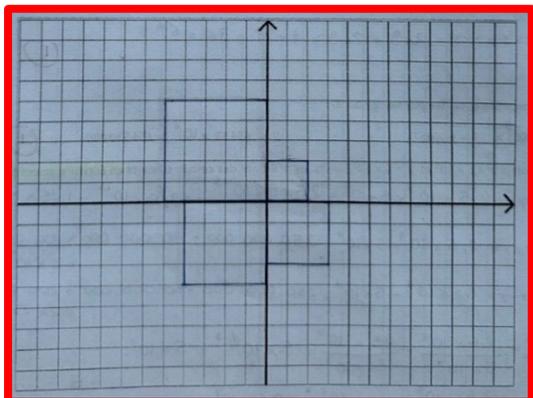
- La figure de base, le carré est reproduit.
- Le nombre de répétitions est identifié.
- Le rôle de la variable Longueur ainsi que sa valeur sont compris.
- L'utilisation du bloc est acquise.

DONC : Le principe de l'algorithme est acquis pour la plupart.

La difficulté majeure est l'orientation du lutin, et en découle sa position. Le repérage en scratch est complété par l'orientation du lutin. C'est une donnée nouvelle par rapport au cours de mathématiques sur le repérage.

Remédiation : travailler sur les blocs « orienter » et « tourner », afin de compléter le repérage du lutin dans la scène.

Productions et Bilan



La différence avec le premier sujet est l'accumulation dans la variable longueur. Les élèves ont bien compris également ce fonctionnement. La difficulté pointée est identique, l'orientation du lutin.

A - OBJECTIFS ATTEINTS :

A l'issue de la séance, les élèves ont compris comment se déplacer, tourner et faire le lien entre grille et pixels.

Les boucles ne posent pas de réels problèmes à comprendre, comme les blocs

B - LIMITES :

Le bloc « S'orienter » pose problème. Il n'est pas intuitif et propre à Scratch. Pas de lien avec le repérage. Le test d'un programme est difficile à mettre en œuvre. Les élèves ne décomposent pas le problème (d'abord faire le motif, puis itérer deux fois, puis plus...)

C - LA SUITE :

Reprendre le bloc s'orienter dans une autre activité
Faire corriger des programmes pour travailler sur le test.

II – ACTIVITES DE TRANSITION DE SCRATCH A PYTHON

ACTIVITE 4: PROGRAMMATION SCRATCH AU COLLEGE/LYCEE D'UNE APPLICATION « SANTE ».

Nom / Etablissement : Olivier PILORGET - Edith BONNERY - Collège Sidney Bechet

Denis RIGUCCINI – Lycée Audiberti

Niveau : 3^e/2^{nde}

Type d'activités/modalités de travail : Travail en groupe en îlots sur tablette sur Scratch. Utilisation de l'ENI de la classe pour la phase de mise en commun. Cette activité a été proposée aux élèves sur une matinée (2h30 – 3h) dans le cadre d'une liaison 3^e/2^{nde}.

Thématique(s) du programme – Objectifs :

Cette activité propose d'élaborer et de programmer « le script d'une application santé ». L'objectif est de déterminer l'Indice de Masse Corporelle d'une personne et d'interpréter le résultat avec les critères de l'OMS (Organisation Mondiale de la Santé) en estimant sa corpulence. Les thèmes mathématiques abordés dans l'activité sont : écrire, mettre au point et exécuter un programme.

Capacités Mathématiques :

Calcul littéral et notion de fonction. Modéliser un problème donné.

Capacités Algorithmique / Informatique :

Blocs d'entrées et de sorties, notions de variables, utiliser les blocs opérateurs pour élaborer un programme de calcul.

PRESENTATION DE L'ACTIVITE

1ere Partie : Découverte du sujet et des documents mis à disposition - 15 min

L'activité commence par une vidéo ludique qui permet en quelques minutes de faire le point avec le groupe classe sur la notion d'IMC.

Les élèves commentent la vidéo, donnent leur avis et proposent des pistes de réflexion pour élaborer une application « santé ».

2e Partie : Travail en groupe de 4 élèves, en débranché - 20 min

L'introduction avec la vidéo permet une mise en activité rapide des élèves. Les notions ont été facilement intégrées par les élèves. Le sujet «application Santé» est fédérateur et permet d'initier des échanges entre tous les membres du groupe :

- A quoi doit ressembler l'application santé ?
- Comment calculer l'IMC d'une personne qui utilise l'application ?
- Comment interpréter le résultat du calcul de l'IMC ?
- Quel retour proposer à l'utilisateur (un bilan ? ; des conseils ? ; un objectif à atteindre ? ; ...)
- Comment programmer cette application sur scratch ?



3e Partie : Phase de programmation en autonomie - 1h30min

Dans chaque groupe, les élèves utilisent les tablettes pour rechercher des informations (élaboration des différentes possibilités pour le retour proposer à l'utilisateur) et ils commencent le script.

- La phase d'élaboration est très riche avec beaucoup d'échanges dans les groupes
- Régulièrement les élèves testent leur application : Le côté script / l'aspect de l'interface de l'application / le retour et les conseils à l'utilisateur.

Point de vigilance : Certains élèves très à l'aise avec Scratch ont tendance à essayer d'aller vite en oubliant l'aspect travail d'équipe.

Les élèves rendent leur travail via l'application Capytale.



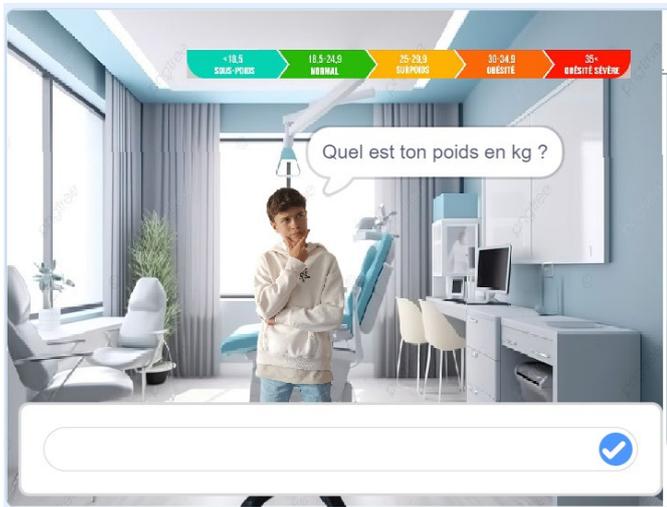
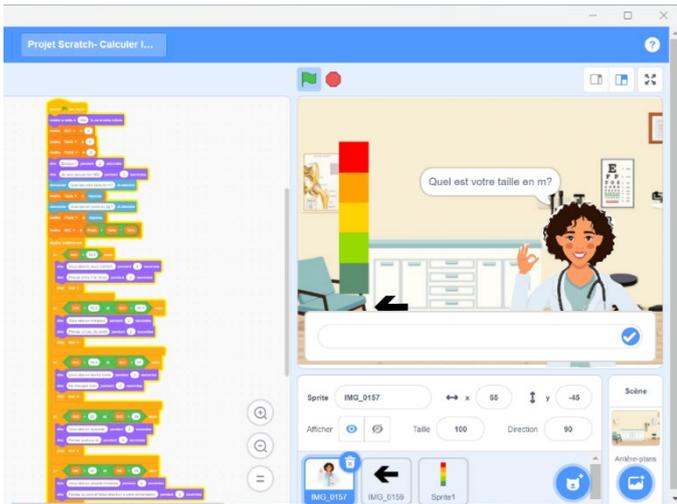
4e Partie : Présentation à la classe

Chaque groupe présente sur l'ENI de la classe son application santé. C'est l'occasion pour eux de mettre en avant les points forts de leur proposition mais également les difficultés rencontrées.

Quelques réflexions d'élèves :

- Le calcul de l'IMC est-il le même pour les filles et pour les garçons ?
- Ne pas s'arrêter sur un objectif de poids à atteindre mais fournir également des conseils pour y arriver (alimentation / Sport / hygiène de vie ...)

EXEMPLES DE PRODUCTIONS D'ÉLÈVES (INTERFACES) :





Tache à prise d'initiatives

« Une appli IMC »

Objectif : A partir des différents documents à disposition, imaginer une application qui, en déterminant l'Indice de Masse Corporelle d'une personne, interprète/analyse le résultat en suivant les critères de l'OMS (Organisation Mondiale de la Santé) en estimant sa corpulence.

- ➔ En plus de faire le point sur l'état actuel de la personne on peut imaginer que l'application donne le nombre de kilos à perdre ou à prendre pour atteindre un « poids idéal ».
- ➔ L'application sera élaborée en utilisant Scratch. L'esthétique de cette application ne doit pas être négligée.
- ➔ La production finale comportera une proposition d'algorithme ainsi qu'une mise en œuvre avec Scratch.

Cette production devra être présentée aux autres élèves à l'oral par l'un des membres de chaque groupe. Elle sera également enregistrée sur l'application Capytale.

doc 1 : Vidéo d'introduction



Source : https://youtu.be/5Vz6cp1RBqs?si=KA5eADIKCvGyQV_r

doc 2 : Critères de l'OMS

Interprétation de l'IMC

IMC (kg·m ⁻²)	Interprétation
moins de 16,5	dénutrition ou famine
16,5 à 18,5	maigre
18,5 à 25	corpulence normale
25 à 30	surpoids
30 à 35	obésité modérée
35 à 40	obésité sévère
plus de 40	obésité morbide ou massive

ACTIVITE 5: PROGRAMMATION SCRATCH AU COLLEGE/LYCEE A TRAVERS LA CONSTRUCTION DE FIGURES GEOMETRIQUES.

Nom / Etablissement : Olivier PILORGET - Edith BONNERY - Collège Sidney Bechet

Denis RIGUCCINI – Lycée Audiberti

Niveau : 3^e/2^{nde}

Type d'activités/modalités de travail : Travail en groupe en îlots sur tablette sur Scratch. Utilisation de l'ENI de la classe pour les phases de mise en commun. Liaison 3^e/2^{nde}

Thématique(s) du programme – Objectifs :

Cette activité propose d'élaborer et de programmer des scripts permettant le tracer d'un certains nombres de frises et de pavages. Elle fait suite à un premier travail en « débranché » sur la construction de figures simples. Les thèmes mathématiques abordés dans l'activité sont : écrire, mettre au point et exécuter un programme.

Capacités Mathématiques :

S'orienter dans le plan et dans l'espace, se repérer, calculer.

Capacités Algorithmique / Informatique :

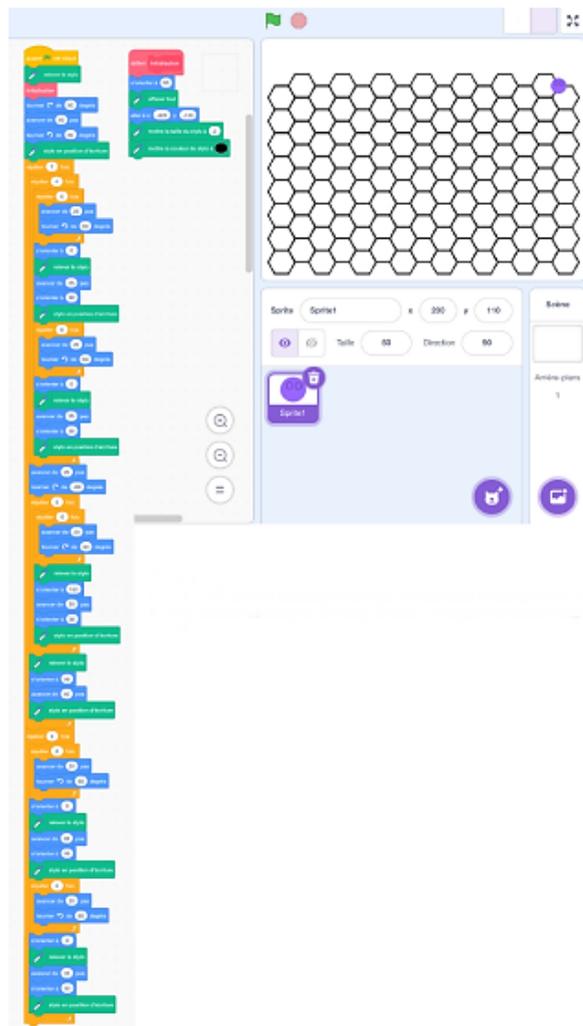
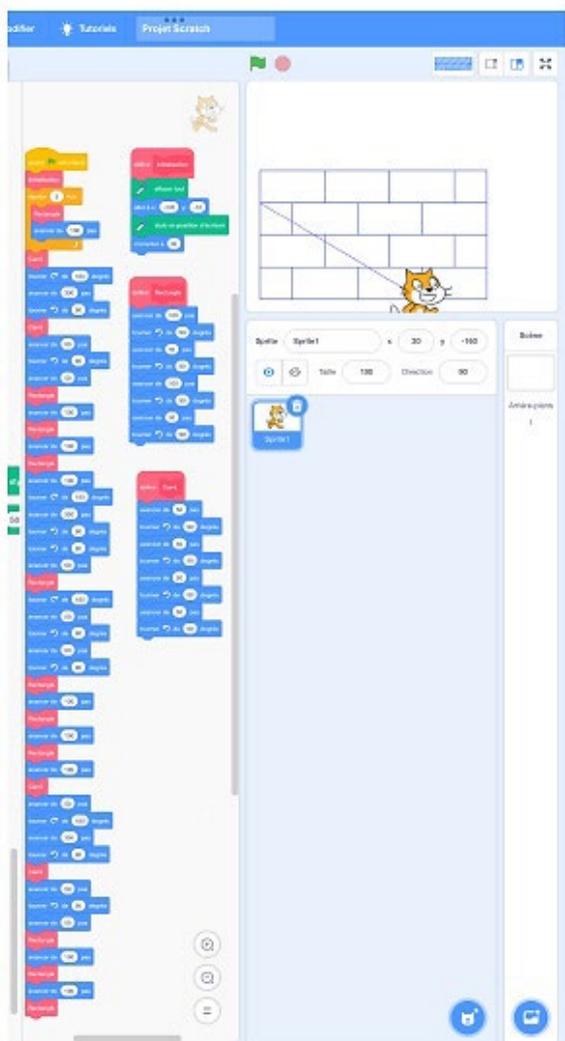
Initialiser, orienter le lutin, mobiliser des blocs (« motifs » ; déplacements), des itérations et des variables.

PRESENTATION DE L'ACTIVITE

L'activité commence par l'élaboration d'une frise et permet après un premier temps en autonomie de faire le point en groupe classe sur les « bonnes » stratégies à mettre en œuvre : conditions d'initialisation, repérage, orientation du lutin et utilisation d'une « figure de base » avec un certain nombre de répétitions. Ce premier temps doit permettre aux élèves d'enchaîner assez rapidement sur l'élaboration de scripts pour réaliser les pavages proposés.

Pour chaque situation, un premier travail d'analyse est proposé. Les différentes figures de bases amènent les élèves à réfléchir sur les stratégies d'élaboration à mettre en œuvre. A chaque fois se pose la question de réussir à aboutir à un script fonctionnel et le plus court possible.

Exemples de scripts « à rallonge » qui permettent, en classe entière, d'échanger sur les bonnes stratégies de programmation à mettre en œuvre :



Les mises en œuvre pour passer d'un étage de motifs à un autre amènent les élèves à travailler la notion de variable et de variable d'accumulation. Les outils mathématiques sont mobilisés pour créer les figures de base les plus complexes et pour déterminer la hauteur des étages dans certains cas (par exemple le théorème de Pythagore dans la question 1.c)).

MODALITES DE MISE EN PLACE

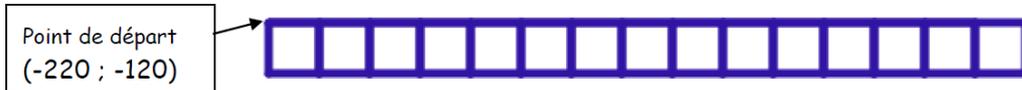
Chaque groupe de 4 élèves dispose de seulement deux tablettes pour favoriser les échanges et faciliter la mise en place d'un réel travail d'équipe. Des points d'étape réguliers sont effectués en classe entière en partageant les idées des différents groupes via l'ENI de la classe.



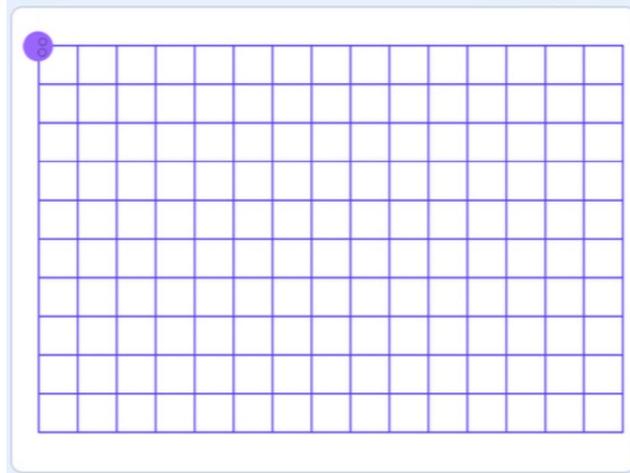
Activité : Frises et pavages

De Scratch à Python

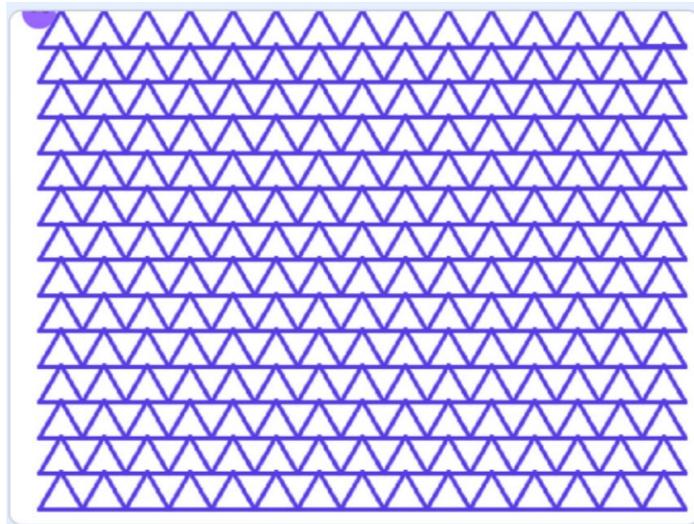
1. a. Sur scratch réaliser une frise de 15 carrés de 30 pas de côté de la couleur de ton choix :



- b. On va maintenant répéter la frise pour obtenir un pavage de toute la scène :



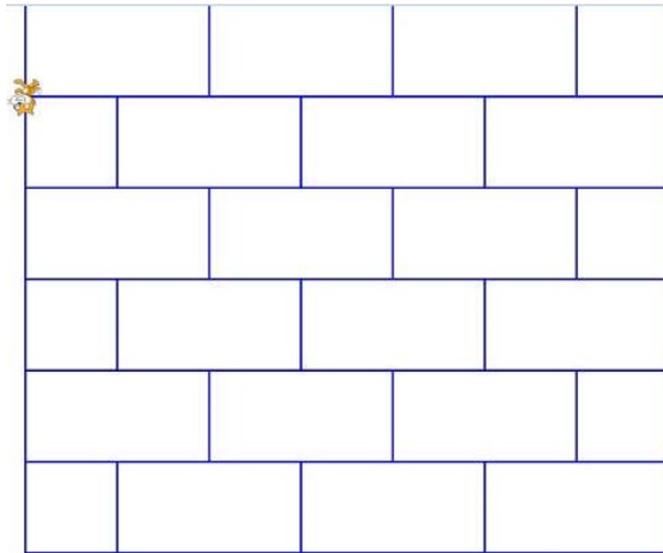
- c. En s'inspirant de la méthode précédente, réaliser un pavage avec des triangles équilatéraux de 30 pas de côté :



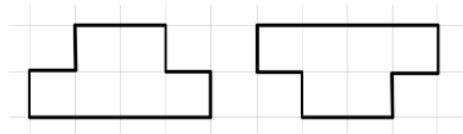
2. Tracer un mur constitué de briques carrés (50 par 50) et de briques rectangulaires (50 par 100)

Pour simplifier l'écriture du script on pourra créer un bloc « Carré » et un bloc « Rectangle » ...

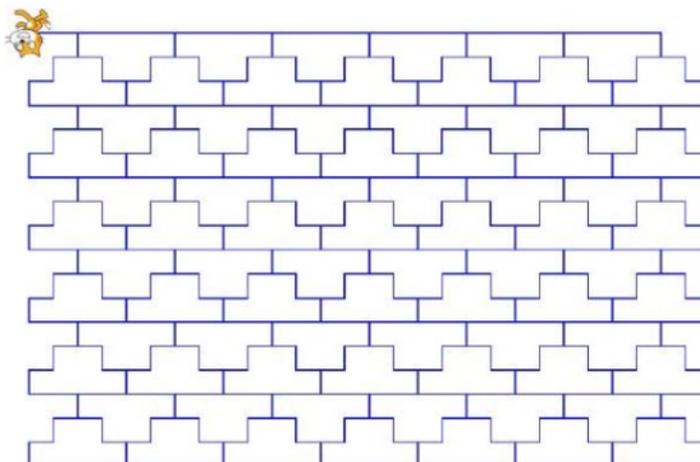




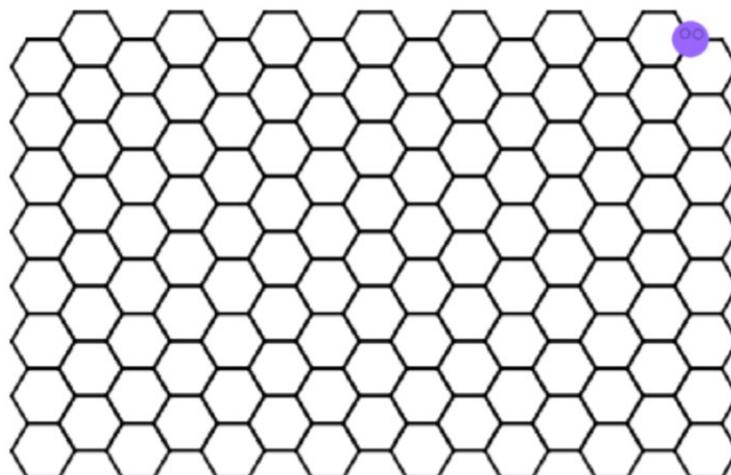
3. On considère les deux figures ci-contre :



En utilisant ces deux figures avec des dimensions adaptées, réaliser le pavage ci-dessous :



4. Réaliser le pavage ci-dessous constitué d'hexagones réguliers :



ACTIVITE DE SCRATCH A PYTHON L'algorithmique du collège au lycée

ACTIVITE 6: INTRODUCTION DU LANGAGE PYTHON EN CLASSE DE SECONDE

Nom / Etablissement : Françoise HELSON – Cédric GOURJON – Lycée S.Veil à Valbonne

Niveau : 3^e/2nde

Type d'activités/modalités de travail : Travail en groupe en îlots sur tablette sur un notebook jupyter (Capytale)

Lien vers l'activité : <https://capytale2.ac-paris.fr/web/c/5c2d-3671142>

Thématique(s) du programme – Objectifs :

L'exercice qui sert de support est un programme de calcul du même type que ceux qui sont vus au collège. Les thèmes mathématiques du programme de seconde abordés dans l'activité sont la manipulation d'expressions littérales et celles de fonctions en langage python.

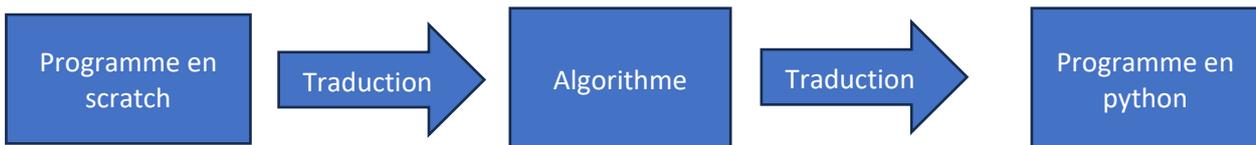
Capacités Mathématiques :

- Choisir la forme la plus adaptée (factorisée, développée réduite) d'une expression en vue de la résolution d'un problème.

Capacités Algorithmique / Informatique :

- Écrire des fonctions simples ; lire, comprendre, modifier, compléter des fonctions plus complexes. Appeler une fonction.

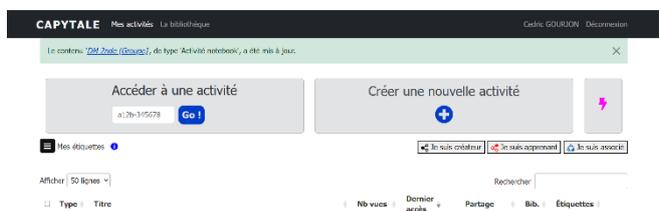
Pour faciliter le passage de scratch à python, l'activité oblige l'élève à repasser par un algorithme. Le principe est donc le suivant :



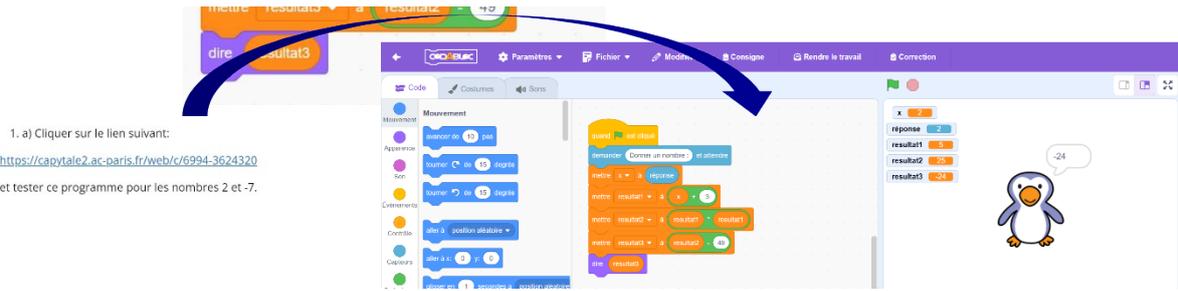
La traduction directe de scratch en python n'est pas naturel pour l'élève et ajoute de la complexité au niveau de l'apprentissage du langage python. L'objectif de l'activité est non pas de faire traduire directement le programme scratch en python mais de le traduire dans un premier temps en un algorithme, compétence déjà travaillée au collège puis de traduire l'algorithme en langage python sous forme d'une fonction. L'objectif à ce stade de l'activité est de familiariser l'élève avec les commandes python (def, return, etc.) en lui faisant compléter des programmes python préconstruits à partir de l'algorithme.

PRESENTATION DE L'ACTIVITE

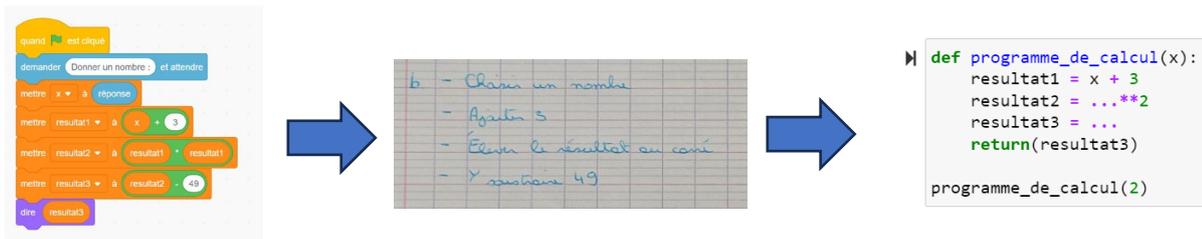
L'activité est un notebook jupyter créé à partir de la plateforme Capytale disponible en collège comme en lycée.



Pour faciliter la compréhension du programme scratch, un lien a été placé sur le notebook pour permettre à l'élève d'accéder à codabloc pour tester le programme avec différentes valeurs.



La première partie de l'activité consiste à tester le programme de calcul, écrire l'algorithme de calcul correspondant puis de créer la fonction python programme_de_calcul construite à partir de l'algorithme.



En parallèle, l'élève est amené à déterminer successivement, l'expression littérale correspondante, son expression développée puis son expression factorisée et ainsi travailler sur la transformation et la manipulation d'expressions littérales.

Pour chacune de ces formes obtenues par l'élève, une nouvelle fonction python est créée pour qu'il puisse tester des valeurs et vérifier la justesse de son résultat.

4) b)
On souhaite écrire une deuxième version du programme de calcul mais cette fois-ci avec l'expression factorisée trouvée à la question 4.a).

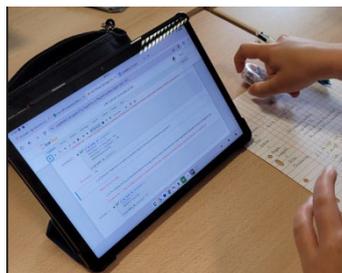
Compléter les "..." dans le programme écrit ci-dessous et l'exécuter. Comparer le résultat obtenu avec votre réponse à la question 2)

```
def programme_de_calcul2(x):
    resultat = ...
    return(resultat)

programme_de_calcul2(-10)
```

MODALITES DE LA MISE EN PLACE

Chaque élève a rendu une copie sur laquelle figurait le travail sur les expressions et un notebook avec toutes les fonctions python programmées.



Travail en îlots sur les tablettes sur Capytale disponible sur atrium en lycée

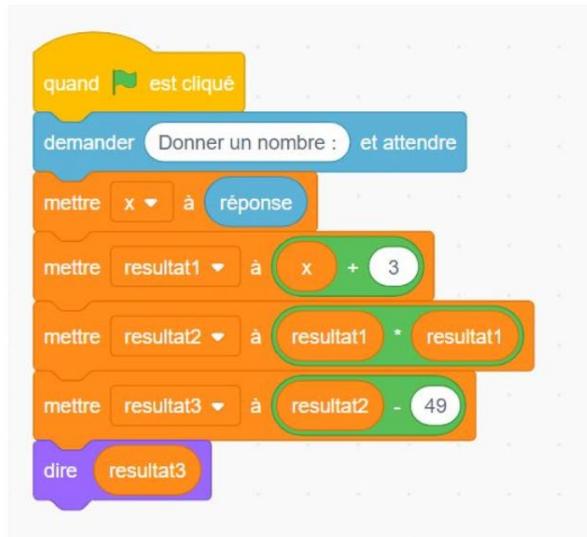
Devoir de groupe n°2 : Scratch à Python

Seconde

Noms et prénoms des membres du groupe :

Ecrire les noms des membres du groupe ici

On considère le programme de calcul suivant :



1) a)

Cliquer sur le lien suivant: <https://capytale2.ac-paris.fr/web/c/6994-3624320> et tester ce programme pour les nombres 2 et -7. Retrouver ces résultats par le calcul.

Répondre sur la copie

1) b)

Compléter l'algorithme de calcul correspondant au programme scratch ci-dessus :

Choisir un nombre

Ajouter 3

...

Répondre sur la copie

2)

Pour quelle(s) valeur(s) choisie(s) le programme donne un résultat nul.

Répondre sur la copie

3) a)

Quelle expression est obtenue en fin de programme en choisissant au départ un nombre x quelconque?

Répondre sur la copie

3) b)

On souhaite écrire un programme en langage python qui à partir d'un nombre x choisi renvoie le résultat donné par le programme de calcul ci-dessus.

Compléter les "..." dans le programme écrit ci-dessous et l'exécuter. Comparer le résultat obtenu avec la valeur trouvée à la question 1)

```
Entrée[ ]: ▶ def programme_de_calcul(x):  
    resultat1 = x + 3  
    resultat2 = ...**2  
    resultat3 = ...  
    return(resultat3)  
  
programme_de_calcul(2)
```

Cliquer ici pour obtenir une explication sur la programmation des fonctions en python

4) a)

Factoriser l'expression trouvée à la question 3)a) et retrouver le résultat de la question 2.

Répondre sur la copie

4) b)

On souhaite écrire une deuxième version du programme de calcul mais cette fois-ci avec l'expression factorisée trouvée à la question 4.a).

Compléter les "..." dans le programme écrit ci-dessous et l'exécuter. Comparer le résultat obtenu avec votre réponse à la question 2)

```
Entrée[ ]: ▶ def programme_de_calcul2(x):  
    resultat = ...  
    return(resultat)  
  
programme_de_calcul2(-10)
```

5) a)

Développer l'expression trouvée à la question 3)a).

Répondre sur la copie

5) b)

On souhaite écrire une troisième version du programme de calcul de la question 1 mais cette fois-ci avec l'expression développée trouvée à la question 5)a).

Compléter les "..." dans le programme écrit ci-dessous et l'exécuter. Comparer les résultats obtenus avec votre réponse à la question 1)

```
Entrée[ ]: ▶ def programme_de_calcul3(x):  
    resultat = ...  
    return(resultat)  
  
programme_de_calcul3(2)
```

```
Entrée[ ]: ▶ programme_de_calcul3(-7)
```

```
Entrée[ ]: ▶ def nom_fonction(parametre):  
    ...  
    ...  
    return(resultat)
```

III – ACTIVITES DE PROGRAMMATION EN PYTHON AU LYCEE

ACTIVITE 7: ACTIVITE SUR LA DETERMINATION D'UN EXTREMUM EN CLASSE DE SECONDE

Nom / Etablissement : Saloua RHIOUI – Emma GILESTRI – Lycée S.Veil à Valbonne

Niveau : 2nde

Type d'activités/modalités de travail : travail individuel sur 1h à terminer si besoin à la maison (en binôme ou seul) sur un notebook jupyter (Capytale)

Lien vers l'activité : <https://capytale2.ac-paris.fr/web/c/ea1b-2854902>

Thématique(s) du programme – Objectifs :

L'objectif est de créer un programme Python en utilisant les boucles bornées afin de déterminer les valeurs minimales et maximales d'une fonction sur un intervalle

Capacités Mathématiques :

-Déterminer les images d'une fonction ; la valeur minimale et maximale d'une fonction sur un interval donné

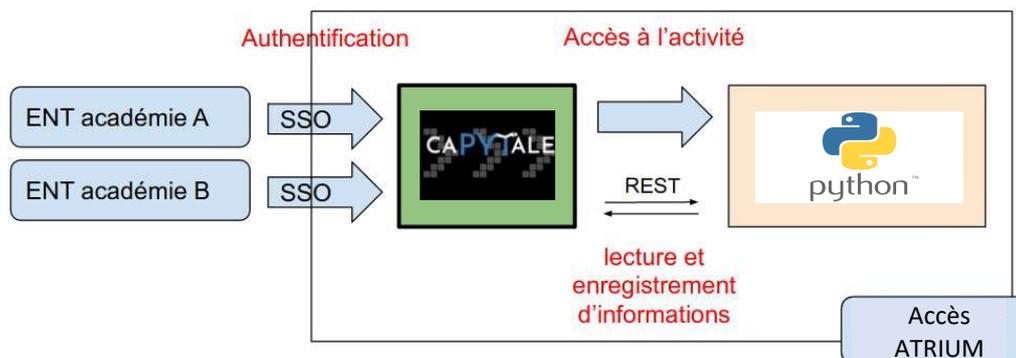
Capacités Algorithmique / Informatique :

-définir des fonctions simples ; lire, comprendre, modifier, compléter des scripts et appeler une fonction. Introduire la boucle for et les conditions.

PRESENTATION DE L'ACTIVITE :

L'activité est un notebook Jupyter créé à partir de la plateforme Capytale disponible en collège comme en lycée. L'activité intervient une fois le chapitre sur les fonctions terminé rappelle sur l'utilisation d'une boucle bornée puis l'application au chapitre des fonctions.

Le fonctionnement du partage est illustré dans le schéma ci-dessous.



Avantages de Capytale :

- Accès rapide depuis Atrium
- La Centralisation de travail (suivi en temps réel, correction rapide et retour des copies aux élèves)
- Le partage des ressources avec les collègues de la discipline

PRESENTATION DU TP

Travail de l'élève:

Appréciation et notation

Résultat d'exécution

TP fonction Rafael DELATTRE — 2nde04

Appréciation: attention la précision de la question 9 est au centième près

Évaluation: bien

Rappel Boucle bornée (boucle for)

Les programmes ci-dessous sont des exemples simples d'utilisation de la boucle `for`.

Exemple 1: On répète 5 fois l'instruction `print("Bonjour")`.
Le bloc d'instructions à répéter doit être **indenté**, c'est-à-dire décalé de quelques espaces vers la droite.
La variable `i` est l'indice de boucle (ou variable de boucle ou compteur de boucle). On peut lui donner le nom que l'on veut.
La fonction `range()` permet de créer des intervalles. On utilisera `range(a,b)` pour créer l'intervalle des entiers successifs compris entre `a`(inclus) et `b`(exclu).

Entrée[3]:

```
#exemple 1:
for i in range(0,5):
    print("Bonjour", i)
```

Bonjour 0
Bonjour 1
Bonjour 2
Bonjour 3
Bonjour 4

Entrée[4]:

```
#exemple 2: on repète plusieurs fois "Bonjour" pour tout entier i qui appartient à l'intervalle [2,6]
for i in range(2, 6) :
    print("Bonjour", i)
#fin de la boucle for
print("à la fin de la boucle for i=", i)
```

Bonjour 2
Bonjour 3
Bonjour 4
Bonjour 5
à la fin de la boucle for i= 5

TP Maths Python

Chapitre des fonctions Le but du TP est de créer un programme Python pour :

- calculer les images par la fonction f pour des valeurs de x dans un intervalle donné,
- et déterminer les valeurs minimales et maximales de la fonction sur cet intervalle, en affichant les résultats.

On considère par la suite la fonction f définie par:

$$f(x) = x^2 - 3x + 2 \text{ sur un intervalle donné } [a;b]$$

Lire et répondre aux questions suivantes:

Question 1 : Rappel

Définir en Python la fonction f .

```
Entrée[ ]: # Rappel : Définition d'une fonction
def Nom_de_fonction(variable):
    resultat = expression_littéral_de_la_fonction
    return resultat
```

```
Entrée[ ]: # Définition de la fonction à étudier
def ...(...):
    .....
    return .....
```

Question 2 : On considère par la suite que la fonction f définie pour x compris entre -5 et 5 .
On note $[a;b]$ l'intervalle tel que $a=-5$ et $b=5$.

Modifier le programme Python ci-dessous pour calculer les images des entiers relatifs compris entre -5 et 5 .

```
Entrée[ ] : # Initialisation des bornes de l'intervalle d'étude
a = ...
b = ...
# Boucle pour parcourir l'intervalle entre -5 à 5
for x in range(a, b + 1):
    y = .....
    # Affichage des résultats
    print("L'image de x=",x," par la fonction f est : ",y)
```

Question 3 : Que représentent a et b dans le programme?

Question 4 : A partir du code précédent compléter le tableau suivant :

x	-5	-4	-3	-2	-1	0	1	2	3	4	5
f(x)											

Question 5 : Que fait l'instruction "for" ?

Question 6 : D'après le tableau de la question 4 déterminer la valeur minimale

Question 7 : Tester le programme ci-dessous et déterminer son rôle en complétant les commentaires

```
Entrée[ ] : # Initialisation des valeurs minimales
min_value = f(a)
# Boucle pour parcourir l'intervalle et .....
for x in range(a, b + 1):
    y = f(x)
    if y < min_value:# .....
        min_value = y # .....
# Affichage des résultats
print("Le minimum de la fonction sur l'intervalle [{}, {}] est : {}".format(a, b, min_value))
```

L'activité possède 9 questions :

- pour atteindre les objectifs de TP il est nécessaire d'aller jusqu'à la Q7.
- **Q 8:** des applications directes de la Q7 en utilisant d'autres fonctions ,
- **Q 9:** Pour aller plus loin et avoir plus de précision sur la détermination des extremums

Question 8 : Modifier le précédent code pour trouver la valeur maximale de la fonction f sur l'intervalle [-10;12].

Entrée[] :

APPLICATION :Déterminer à l'aide d'un programme Python les extremums des fonctions définies ci-dessous :

- $g(x)=x^3-x^2-5$ pour x entre 1 et 1000 ,
- $h(x)=x^5-2x^3+3$ pour x entre -200 et 0

Le Programme Python pour déterminer les extremums de la fonction g:

Entrée[] :

Le Programme Python pour déterminer les extremums de la fonction h:

Entrée[] :

POUR ALLER PLUS LOIN:

Question 9 :

On souhaite obtenir plus de précision sur la valeur du minimum de la fonction initiale f. Modifier le programme de la question 7 afin d'affiner la précision (au centième près).

Entrée[] :

DEROULE DE LA SEANCE

Etape 0 :

Questions flash sur les notions nécessaires en lien avec le chapitre des fonctions

Etape 1 :

Sur Capytale: début de l'activité avec des rappels sur l'utilisation d'une boucle bornée (2 exemples et une application)

Etape 2 :

Suite activité Capytale avec application au chapitre des fonctions :

- Définir une fonction, son intervalle, calculs d'images
- Déterminer les valeurs minimales et maximales d'une fonction sur un intervalle donné

Etape 3 :

Pour aller plus loin : créer un programme pour affiner le pas dans l'intervalle

PRODUCTIONS D'ELEVES

Question 7 : Tester le programme ci-dessous et déterminer son rôle en complétant les commentaires

```
# Initialisation des valeurs minimales
def f(a):
    min_value = f(a)
# Boucle pour parcourir l'intervalle et la valeur minimale de f
for x in range(a, b + 1):
    y = f(x)
    if y < min_value:# f(x)<f(a)
        min_value = y # f(a)=f(x)
# Affichage des résultats
print("Le minimum de la fonction sur l'intervalle [{}, {}] est : {}".format(a, b, min_value))
```

```
Traceback (most recent call last):
  File "<input>", line 5, in <module>
NameError: name 'a' is not defined
```

APPLICATION :Déterminer à l'aide d'un programme Python les extremums des fonctions définies ci-dessous

- $g(x)=x^3-x^2-5$ pour x entre 1 et 1000,
- $h(x)=x^5-2x^3+3$ pour x entre -200 et 0

Le Programme Python pour déterminer les extremums de la fonction g:

```
def g(x):
    resultat=x*3-x**2-5
    return resultat
a= 1
b= 1000
max_value = g(b)
for x in range(a, b + 1):
    y = f(x)
    if y > max_value:
        max_value = y
print("Le maximum de la fonction sur l'intervalle [{}, {}] est : {}".format(a, b, max_value))
    if y < min_value:
        min_value=y
print("Le minimum de la fonction sur l'intervalle [{}, {}] est : {}".format(a, b, min_value))
```

```
File "<input>", line 12
    if y < min_value:
IndentationError: unexpected indent
```

Le Programme Python pour déterminer les extremums de la fonction h:

ERREURS FREQUENTES

- Ajouts d'instructions inutiles
- Mauvaise indentation
- Utilisation des variables sans les définir et les initialiser
- Exécuter chacune des questions
- Les étudiants ont du mal à comprendre le lien entre les questions.

Exemple : la fonction f est définie dans la Q1 et est utilisée dans la Q

PISTES D'AMELIORATIONS

1) Les notions de programmation telles que :

- définition d'une fonction avec l'instruction def
- boucle bornée et son formalisme
- recherche d'un minimum par comparaison successives

auraient pu faire l'objet d'une séance en amont

2) Mise en commun et correction ensemble une fois la moitié de l'activité réalisée

BILAN

Lors du test sur le demi-groupe le plus autonome, identification de l'importance des rappels sur les notions utilisées en programmation (ajout de la partie rappel dans l'activité Capytale)

Globalement bien passé, la plupart des élèves ont pu aller jusqu'à l'application

Mais le groupe ayant plus de difficulté aurait eu besoin de plus de temps.

ACTIVITE DE SCRATCH A PYTHON L'algorithmique du collège au lycée

ACTIVITE 8 : VARIATIONS ET NOTION DE LIMITES D'UNE SUITE, NIVEAU PREMIERE

Nom du professeur / Etablissement : Frank d'ESQUERMES – Lycée S.Veil à Valbonne

Niveau : Première (Spécialité)

Type d'activités/modalités de travail : TP pouvant être fait seul ou en groupe sur tablette sur un notebook Jupyter (Capytale)

Lien vers l'activité : <https://capytale2.ac-paris.fr/web/c/5c2d-3671142>

Thématique(s) du programme – Objectifs :

Le niveau de Première sous-entend une connaissance des bases du langage Python, tout en introduisant des éléments de langage pour les graphiques.

Capacités Mathématiques :

- Sens de variation d'une suite.
- Sur des exemples, introduction intuitive de la notion de limite, finie ou infinie, d'une suite.

Capacités Algorithmique / Informatique :

- Générer une liste (en extension, par ajouts successifs ou en compréhension).
- Manipuler des éléments d'une liste (ajouter, supprimer...) et leurs indices.
- Parcourir une liste.
- Itérer sur les éléments d'une liste.

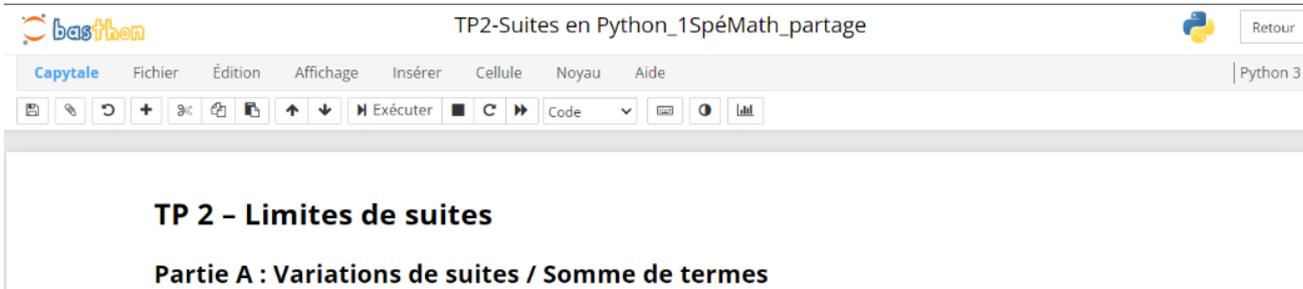
Les élèves auront déjà travaillé les listes dans un TP précédent. Le travail est aussi bien sur les connaissances et l'application du cours que son utilisation en programmation en langage Python.

Les élèves sont guidés, et aidés sur les nouvelles notions, que ce soit en programmation qu'en connaissance, comme pour la représentation graphique d'une suite en Python, ou la notion de limite.

On s'appuie sur des exemples connus, comme le problème de l'échiquier, ou Fibonacci, en passant par des problèmes plus « terre à terre » de modélisation. Une difficulté est l'utilisation de paramètres.

PRESENTATION DE L'ACTIVITE

L'activité est un notebook Jupyter créé à partir de la plateforme Capytale disponible en collège comme en lycée.



La première partie de l'activité consiste à l'étude des variations d'une suite connue comme la légende de Sissa. On alterne des questions classiques et des questions de programmation.

a) Définissez la suite r_n , pour n entier naturel non nul et inférieur ou égal à 64, qui donne le nombre de grains de riz sur la n -ième case de l'échiquier.

réponse :

b) Écrire une fonction `suite_r(n)` qui affiche les n premiers termes de la suite r_n dans une liste ($1 \leq n \leq 64$).

Entrée[]:

c) Conjecturer alors le sens de variation de la suite (r_n) .

réponse :

d) Écrire une fonction `variation_suite_r()` qui affiche les rapports successifs $\frac{r_{n+1}}{r_n}$ dans une liste ($1 \leq n \leq 63$) et teste si ces rapports sont tous supérieurs à 1.

Entrée[]:

e) Enfin, écrire une fonction `somme_suite_r()` qui affiche le nombre total de grain de riz sur l'échiquier.

Entrée[]:

Entrée[]: `somme_suite_r()`

f) Conclure quant à la phrase du conseiller (vous pouvez vous aider d'une recherche sur le web que la production mondiale de riz et la masse d'un grain de riz moyen).

Les élèves ne sont pas trop guidés dans cette partie, car elle fait appel à des connaissances de programmation déjà abordées.

Cette première partie du TP se poursuit par l'étude d'une suite à paramètres, permettant de comprendre les influences de certains paramètres sur les variations d'une suite.

2. Un nombre d'abonnés fluctuant....

Un site de location de films en ligne possède 30 000 abonnés la première année. Chaque année, il perd 50% de ses abonnés, mais en gagne P milliers de nouveaux ($0 < P \leq 30$).

a) Écrire une fonction `suite_l(P,n)` qui affiche les n premiers termes de la suite I_n , le nombre de milliers d'abonnés au bout de n années, dans une liste.

Entrée[]: `def suite_l(P,n):`
`...`

b) Écrire une fonction `variation_suite_l(P,n)` qui affiche les différences successives $I_{n+1} - I_n$ dans une liste pour $n \geq 1$ (n est choisi par l'utilisateur) et teste le signe de la différence, en prenant comme valeur $P = 14$ et $n = 17$ pour l'exécuter la première fois. En déduire une conjecture sur le sens de variation de (I_n) pour $P = 14$.

Entrée[]: `def variation_suite_l(P,n):`
`...`

Entrée[]: `variation_suite_l(14,17)`

conjecture :

c) Tester avec $P = 16$, puis $P = 15$ pour $n = 17$; donner dans chacun de ces cas une conjecture sur le sens de variation de (I_n) .

Entrée[]: `variation_suite_l(16,17)`

Entrée[]: `variation_suite_l(15,17)`

conjectures :

Pour la seconde partie, les élèves font connaissance de la librairie *matplotlib*, et sont guidés cette fois pour la programmation.

Partie B : Notions de limite de suites

1. En Python....

Pour tracer un graphique, on peut utiliser les librairies *matplotlib* et *pyplot*; une fois installées, on met la ligne :

```
from matplotlib import pyplot as plt
```

en introduction du programme Python. Pour placer un point de coordonnées (a,b) , on utilise la commande :

```
plt.plot(a,b, marker="o", color="red")
```

qui donne un point marqué « o » de couleur rouge.

2. Repeuplement

Dans un village, le maire a un politique de repeuplement qu'il doit défendre : le nombre d'habitants doit passer de 650 habitants à 1 000 en maximum 10 ans.

Il a observé sur les 30 dernières années que le village perd 10% de sa population chaque année, mais que 118 personnes en moyenne emménagent en contrepartie.

a) En notant (h_n) la suite du nombre d'habitant au bout de n années, exprimer cette suite par récurrence.

réponse :

b) Écrire une fonction `terme_suite_h(n)` qui affiche le terme d'indice n de la suite h_n .

Entrée[]: ▶

c) Écrire une fonction `graph_suite_h(n)` qui affiche les termes d'indice jusqu'à n de la suite (h_n) dans un graphe. Tester le programme pour $n = 10$. Conjecturer le sens de variation de la suite (h_n) .

Entrée[]: ▶

```
from matplotlib import pyplot as plt
def graph_suite_h(n):
    ...
```

L'activité se termine par la suite de Fibonacci, après avoir repris la suite de la première partie sur la location de film, afin de bien comprendre la notion de limite.

MODALITES DE LA MISE EN PLACE

Les élèves ont déjà travaillé le TP1 pour avoir les notions Python des listes, ils travaillent donc sur leur tablette, seul ou en petits groupes de 2-3 élèves ; ayant plusieurs classes de Spécialité, les deux situations ont été mises en place, avec des résultats similaires, car la réussite vient surtout de la capacité à avoir assimiler les listes lors du premier TP, et la volonté d'apprendre le langage.

C'est donc au professeur de s'adapter selon le niveau de programmation de ses élèves, les deux possibilités fonctionnant plutôt bien avec ce distinguo.

TP 2 – Limites de suites

Partie A : Variations de suites / Somme de termes

1. Une légende...

On raconte que le roi Belkib (Indes) promet une récompense fabuleuse à qui lui proposerait une distraction qui le satisferait. Lorsque le sage Sissa, fils du Brahmine Dahir, lui présenta le jeu d'échecs, le souverain, demanda à Sissa ce que celui-ci souhaitait en échange de ce cadeau extraordinaire. Sissa demanda au prince de déposer un grain de riz sur la première case, deux sur la deuxième, quatre sur la troisième, et ainsi de suite pour remplir l'échiquier en doublant la quantité de grain à chaque case. Le prince accorda immédiatement cette récompense sans se douter de ce qui allait suivre. Son conseiller lui expliqua qu'il venait de précipiter le royaume dans la ruine car les récoltes de l'année ne suffiraient pas à payer Sissa.

a) Définissez la suite r_n , pour n entier naturel non nul et inférieur ou égal à 64, qui donne le nombre de grains de riz sur la n -ième case de l'échiquier.

réponse :

b) Écrire une fonction `suite_r(n)` qui affiche les n premiers termes de la suite r_n dans une liste ($1 \leq n \leq 64$).

Entrée[]:

c) Conjecturer alors le sens de variation de la suite (r_n) .

réponse :

d) Écrire une fonction `variation_suite_r()` qui affiche les rapports successifs $\frac{r_{n+1}}{r_n}$ dans une liste ($1 \leq n \leq 63$) et teste si ces rapports sont tous supérieurs à 1.

Entrée[]:

e) Enfin, écrire une fonction `somme_suite_r()` qui affiche le nombre total de grain de riz sur l'échiquier.

Entrée[]:

Entrée[]: `somme_suite_r()`

f) Conclure quant à la phrase du conseiller (vous pouvez vous aider d'une recherche sur le web que la production mondiale de riz et la masse d'un grain de riz moyen).

réponse :

2. Un nombre d'abonnés fluctuant...

Un site de location de films en ligne possède 30 000 abonnés la première année. Chaque année, il perd 50% de ses abonnés, mais en gagne P milliers de nouveaux ($0 < P \leq 30$).

a) Écrire une fonction `suite_l(P,n)` qui affiche les n premiers termes de la suite l_n , le nombre de milliers d'abonnés au bout de n années, dans une liste.

Entrée[]: `def suite_l(P,n):`
`...`

b) Écrire une fonction `variation_suite_l(P,n)` qui affiche les différences successives $l_{n+1} - l_n$ dans une liste pour $n \geq 1$ (n est choisi par l'utilisateur) et teste le signe de la différence, en prenant comme valeur $P = 14$ et $n = 17$ pour l'exécuter la première fois. En déduire une conjecture sur le sens de variation de (l_n) pour $P = 14$.

Entrée[]: `def variation_suite_l(P,n):`
`...`

Entrée[]: `variation_suite_l(14,17)`

conjecture :

c) Tester avec $P = 16$, puis $P = 15$ pour $n = 17$; donner dans chacun de ces cas une conjecture sur le sens de variation de (l_n) .

Entrée[]: `variation_suite_l(16,17)`

Entrée[]: `variation_suite_l(15,17)`

conjectures :

3. Pour aller plus loin...

a) Écrire une fonction `suite_g(i,q,n)` qui calcule les n premiers termes d'une suite géométrique (g_n) de premier terme $g_0 = i$ et de raison q .

Entrée[]:

b) Écrire une fonction `variation_suite_g(i,q,n)` qui affiche les rapports successifs $\frac{g_{n+1}}{g_n}$ dans une liste et les différences successives $g_{n+1} - g_n$ dans une autre liste pour $n \geq 1$ (n est choisi par l'utilisateur) et affiche si la suite (g_n) est croissante, décroissante, constante ou encore ni croissante, ni décroissante.

Entrée[]:

Partie B : Notions de limite de suites

1. En Python...

Pour tracer un graphique, on peut utiliser les bibliothèques `matplotlib` et `pyplot`; une fois installées, on met la ligne :

```
from matplotlib import pyplot as plt
```

en introduction du programme Python. Pour placer un point de coordonnées (a,b) , on utilise la commande :

```
plt.plot(a,b, marker="o", color="red")
```

qui donne un point marqué « o » de couleur rouge.

2. Repeuplement

Dans un village, le maire a un politique de repeuplement qu'il doit défendre : le nombre d'habitants doit passer de 650 habitants à 1 000 en maximum 10 ans.

Il a observé sur les 30 dernières années que le village perd 10% de sa population chaque année, mais que 118 personnes en moyenne emménagent en contrepartie.

a) En notant (h_n) la suite du nombre d'habitant au bout de n années, exprimer cette suite par récurrence.

réponse :

b) Écrire une fonction `terme_suite_h(n)` qui affiche le terme d'indice n de la suite h_n .

Entrée[]:

c) Écrire une fonction `graph_suite_h(n)` qui affiche les termes d'indice jusqu'à n de la suite (h_n) dans un graphe. Tester le programme pour $n = 10$. Conjecturer le sens de variation de la suite (h_n) .

```
Entrée[ ]: ▶ from matplotlib import pyplot as plt
def graph_suite_h(n):
    ...
```

```
Entrée[ ]: ▶ graph_suite_h(10)
```

conjecture :

d) À l'aide du graphe obtenu, peut-on dire si le maire atteindra son objectif ? Et avec la fonction `terme_suite_h(10)` ?

```
Entrée[ ]: ▶ terme_suite_h(10)
```

réponses :

e) Le maire veut toutefois savoir si la population après « une longue période » va se stabiliser. Conjecturer une réponse en utilisant `graphe_suite_h(n)` pour n suffisamment grand.

```
Entrée[ ]: ▶
```

conjecture :

3. Et notre site de location de films ?

Notre site de locations films en ligne s'aperçoit que $P = 19$. Il veut connaître son nombre d'abonnés maximal qu'il pourrait atteindre.

a) En utilisant la fonction `suite_l(P,n)`, écrire une fonction `graphe_suite_l(P,n)` qui affiche les termes d'indice jusqu'à n de la suite l_n dans un graphe. Tester le programme pour $n = 20$.

```
Entrée[ ]: ▶ from matplotlib import pyplot as plt
def graphe_suite_l(P,n):
    ...
```

```
Entrée[ ]: ▶ graphe_suite_l(19,20)
```

b) En déduire graphiquement quel sera le nombre d'abonnés maximal que le site pourrait atteindre.

réponse :

4. Et Fibonacci dans tout ça ?

Avec la suite de Fibonacci, écrire une fonction `limite_nbor(n)` qui utilise la fonction `Fibonacci(n)` qui renvoie les rapports successifs des termes de la suite de Fibonacci pour $n \geq 1$ (n est choisi par l'utilisateur). Quelle semble être la limite de ces rapports ? Quel nombre très connu est-ce ?

```
Entrée[ ]: ▶ def limite_nbor(n):
L=[]
...
return(L)
```

```
Entrée[ ]: ▶ limite_nbor(...)
```

réponses :

ACTIVITE DE SCRATCH A PYTHON L'algorithmique du collège au lycée

ACTIVITE 9: UTILISATION DES NOTEBOOKS CAPYTALE POUR 3 ACTIVITES MELANT HISTOIRE DES MATHEMATIQUES ET PROGRAMMATION PYTHON EN TERMINALE.

Nom / Etablissement : Isabelle Mourard - Françoise Helson - Cédric Gourjon – Lycée S.Veil à Valbonne

Niveau : Terminale

Type d'activités/modalités de travail : Travail en groupe en îlots sur tablette sur un notebook jupyter (Capytale)

Lien vers les activités : <https://capytale2.ac-paris.fr/web/c/64fe-706487>

<https://capytale2.ac-paris.fr/web/c/9c91-2165105>

<https://capytale2.ac-paris.fr/web/c/d196-3205805>

Thématique(s) du programme – Objectifs :

La première activité porte sur les coefficients binomiaux et le triangle de Pascal. Le travail proposé a donné lieu à une des premières formalisation de la démonstration par récurrence pour la résolution du cas général du célèbre problème des parties.

La deuxième activité permet d'obtenir une valeur approchée d'une intégrale non accessible au niveau de la classe de terminale avec la méthode de Monte Carlo.

La troisième activité est une étude du flocon de Koch

Capacités Mathématiques :

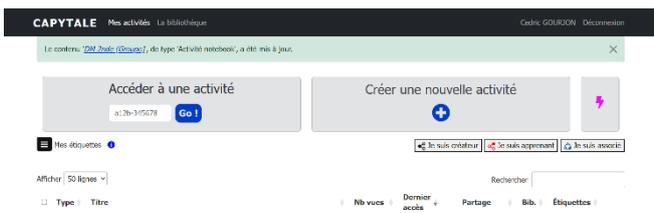
- Pour un entier n donné, génération de la liste des coefficients $\binom{n}{k}$ à l'aide de la relation de Pascal
- Méthode de Monte-Carlo.
- Tous les éléments relatifs au programme de première et de terminale sur les suites numériques

Capacités Algorithmique / Informatique :

- Générer une liste (en extension, par ajouts successifs ou en compréhension).
- Manipuler des éléments d'une liste (ajouter, supprimer...) et leurs indices.
- Parcourir une liste.
- Itérer sur les éléments d'une liste.

PRESENTATION DE LA PREMIERE ACTIVITE

Là encore l'activité est un notebook jupyter créé à partir de la plateforme Capytale.



Devoir maison n°6 (Groupe) : Le triangle de Pascal

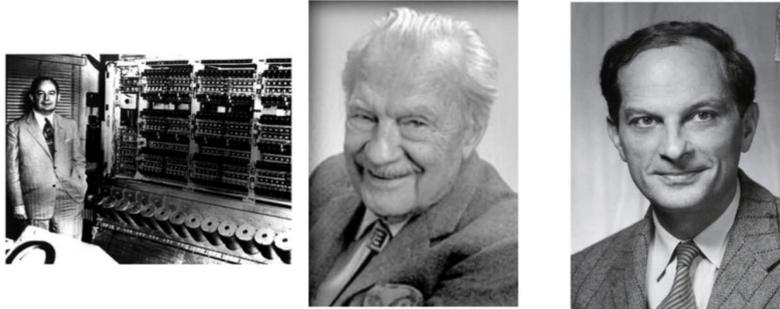
Terminale

Blaise Pascal né le 19 juin 1623 à Clermont en Auvergne et mort le 19 août 1662 à Paris.



PRESENTATION DES DEUX AUTRES ACTIVITES

Pour les deux autres activités, là encore, on trouvera des références historiques qui pourront servir de support au travail sur une question de grand oral. Mais l'intérêt de ces deux activités est qu'elles utilisent ces bibliothèques python qui permettent là aussi d'étoffer les possibilités offertes aux professeurs pour illustrer les notions mises en jeu.



Arthur Von Neumann, Nicholas Metropolis et Stanislaw Ulam

Le Monde

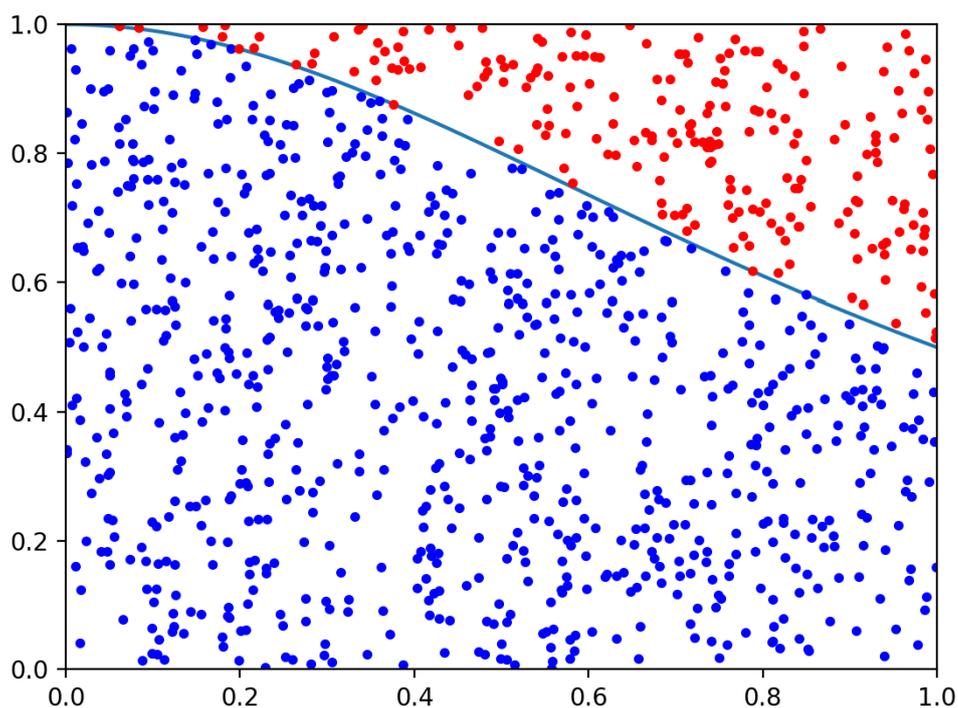
Les mathématiciens jouent à la roulette pour comprendre le hasard

LA LÉGENDE veut que l'idée de faire du hasard un outil de simulation soit née dans l'immédiat après-guerre, au Nouveau-

problème concernant la diffusion des neutrons dans différents matériaux. N'ayant ni le temps ni les moyens d'entreprendre des

Par exemple dans le premier cas, les élèves sont amenés à étudier la méthode de Monte-Carlo à l'aide d'un programme python. C'est la bibliothèque **Matplotlib** qui est utilisée et qui permet de visualiser les points situés « en-dessous » et « au-dessus » de la courbe qui permettront de faire l'estimation en calculant leur fréquence.

Figure 1



Dans la deuxième activité, c'est la bibliothèque **Turtle** qui va permettre le tracé du flocon de Koch.



Helge Von Koch

La fonction récursive python est intégralement donnée, il n'est pas question de faire travailler les élèves sur cette notion qui n'est pas au programme (seulement au programme de NSI) mais l'intérêt est d'illustrer la construction du flocon à chaque étape.

```
from turtle import *

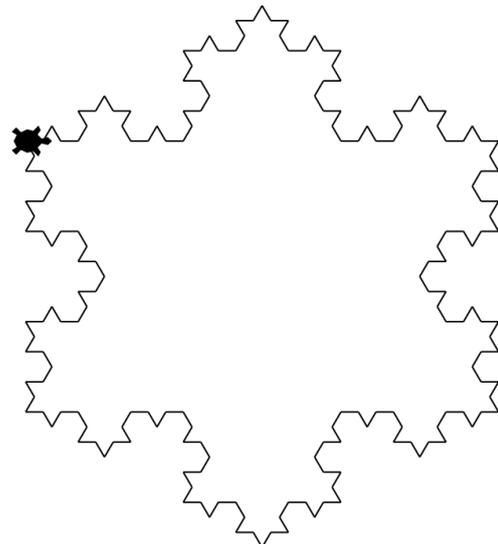
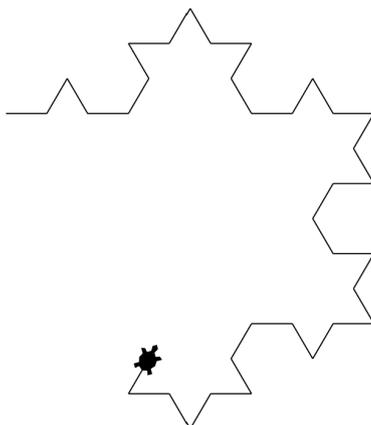
def koch(n,l): #fonction récursive qui permet d'opérer la transformation décrite ci-dessus n fois.
    if n==0:
        down()
        forward(l)
    if n>0:
        down()
        koch(n-1,l/3)
        left(60)
        koch(n-1,l/3)
        right(120)
        koch(n-1,l/3)
        left(60)
        koch(n-1,l/3)

def flocon(n): #fonction qui permet de former le flocon
    for i in range(3):
        koch(n,300)
        right(120)

up()
goto(-300,50)

flocon(3)

mainloop()
```



L'utilisation de la bibliothèque **turtle** n'est pas sans rappeler le déplacement du lutin de Scratch.