

9^e
ANNÉE

PENSONS MATHÉMATIQUES!

**Une approche renouvelée pour l'enseignement
et l'apprentissage des mathématiques**

Minileçon

ALGÈBRE

Établir le lien entre les concepts
algébriques et le codage

RÉSUMÉ

Dans cette minileçon, l'élève utilise le codage pour démontrer sa compréhension de concepts algébriques simples et fait des liens entre la programmation et les concepts de variables, paramètres et équations. L'élève utilise les outils de programmation à des fins de création de graphiques.

PISTES D'OBSERVATION

L'élève :

- › utilise un logiciel de programmation pour créer des tables de valeurs;
- › interprète des relations à l'aide d'un logiciel de programmation;
- › utilise la syntaxe dans un langage de programmation;
- › utilise la boucle et des structures conditionnelles afin de résoudre des problèmes mathématiques;
- › détermine et compare des taux de variation.

MATÉRIEL

- › ordinateur;
 - › logiciel de programmation.
-

CONCEPTS MATHÉMATIQUES

Les concepts mathématiques nommés ci-dessous sont nécessaires pour bien accomplir cette minileçon. Il est important de s'assurer de leur compréhension avant de débiter. Une explication de ceux-ci se trouve dans la section **Concepts mathématiques**.

Domaine d'étude	Concept mathématique
Algèbre	Le lien entre les concepts algébriques et le codage
Algèbre	Les relations linéaires ou non linéaires

Partie 1 – Exploration guidée

Éléments à faire ressortir

- Une boucle est bornée lorsqu'il est possible de prédire exactement le nombre de fois que la boucle va se répéter. Il existe plusieurs façons de borner une boucle.
- En Python, la boucle bornée commence avec le mot clé « for ». Avec Python, les instructions exécutables de la boucle doivent être indentées afin de faire partie de la boucle.



...Instructions qui précèdent la boucle

Boucle bornée →

```
for {variable} in {liste}:  
|  Instruction indentée  
|  Instruction indentée  
...  
else:  
|  Instruction indentée  
|  Instruction indentée  
...
```

La boucle sera exécutée
tant qu'il y a des
éléments dans la liste.

Optionnelle et se
place en dernier.

Instructions qui suivent la boucle...



- Une boucle est non bornée lorsqu'elle dépend d'une condition pour cesser son exécution. Il n'est pas possible de prédire le nombre de fois qu'elle va s'exécuter, il est même possible qu'elle s'exécute à l'infini.



...Instructions qui précèdent la boucle

Boucle non bornée →

```
while {condition}:  
|  Instruction indentée  
|  Instruction indentée  
...
```

La boucle sera exécutée
tant que la condition est
vérifiée.

Instructions qui suivent la boucle...



- Un opérateur de comparaison est utilisé dans une condition afin de vérifier si celle-ci est vraie ou fausse. Il y a 6 opérateurs de comparaison de base, soit :



Opérateur	Définition
<code>==</code> (égal à)	Vérifie si une valeur est égale à une autre.
<code><</code> (plus petit que)	Vérifie si une valeur est strictement plus petite qu'une autre.
<code>></code> (plus grand que)	Vérifie si une valeur est strictement plus grande qu'une autre.
<code><=</code> (plus petit ou égal à)	Vérifie si une valeur est inférieure ou égale à une autre.
<code>>=</code> (plus grand ou égal à)	Vérifie si une valeur est supérieure ou égale à une autre.
<code>!=</code> (pas égale à)	Vérifie si une valeur n'est pas égale à une autre.

- Les structures conditionnelles permettent d'exécuter différentes parties du code selon des conditions spécifiques.
 - La condition `if` (« si »)
 - La condition `if...else` (« si...sinon »)
 - La condition `if...elif...else` (« si...sinon si...sinon »)

Déroulement pour l'exemple 1

- Consulter, au besoin, la fiche **Le lien entre les concepts algébriques et le codage** de la section **Concepts mathématiques** afin de revoir avec les élèves différentes fonctions existantes lors de la programmation d'un code pour répondre à des concepts algébriques ainsi que la terminologie liée à ces concepts en vue de les aider à réaliser l'activité. Dans cette minileçon l'élève découvre la boucle bornée ainsi que les différentes conditions.
- Animer une discussion afin d'activer les connaissances antérieures ou connexes et établir des liens avec des expériences personnelles en posant des questions telles que :
 - Comment les concepts algébriques se prêtent-ils au codage?
- Présenter aux élèves l'exemple 1, soit La création de tables de valeurs pour 2 types de relation.
- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre diverses stratégies pour créer une table de valeurs.
- Consulter les notes pédagogiques fournies dans le corrigé de l'exemple 1 pour obtenir des stratégies d'accompagnement au niveau des processus mathématiques et des éléments de difficulté que les élèves risquent de rencontrer.
- Choisir différentes stratégies de regroupements flexibles afin de répondre aux différents besoins des élèves.

Stratégie de résolution de problèmes pour l'exemple 1

L'élève utilise un logiciel de programmation afin de déterminer les tables de valeurs pour différentes équations.

Notes pédagogiques

Cet exemple permet à l'élève de mettre en application les processus de mathématiques d'établissement des liens et sélection d'outils et de stratégies.

EXEMPLE

1

La création de tables de valeurs pour deux types de relation

À l'aide d'un outil de programmation, crée une table de valeurs pour l'intervalle $-2 \leq x \leq 2$ pour chacune des relations suivantes.

a) $y = 3x + 4$

b) $y = 3x^2 + 4$

Corrigé

Notes pédagogiques

- L'élève sélectionne la stratégie la plus efficace pour résoudre le problème et utilise l'outil de programmation le plus efficace à sa disposition afin d'effectuer différentes tâches mathématiques.
- L'enseignante ou l'enseignant peut offrir à l'élève l'occasion d'utiliser un logiciel de codage par blocs afin de visualiser les étapes de la programmation. De plus, l'enseignante ou l'enseignant pourrait encourager l'utilisation du pseudocode afin de créer une méthodologie du code.
- L'élève peut avoir de la difficulté à comprendre la fonctionnalité de `range()` comparativement à une liste, car afin d'inclure le -2, il faut utiliser le nombre 3 dans la fonction. Ceci est dû à la nature de la fonction `range()` qui arrête à un nombre inférieur à la valeur maximale incluse comme paramètre.
- Afin d'amener l'élève à déterminer la séquence de la suite dans un certain intervalle, il est important de fournir des pistes de réflexion ou un questionnement adéquat tout au long de la résolution du problème.
Voici quelques exemples de questions ou de piste de réflexion afin d'aider l'élève à simplifier des expressions algébriques.

- Quel est l'avantage de l'utilisation de la fonction `range()`?
- Quelle est la stratégie la plus efficace pour un ensemble de nombres compris entre -100 et 100 ?
- Aider l'élève à faire le lien entre l'intervalle de -2 à 2 et l'intervalle créé par la fonction `range()`, et le domaine des équations $y = 3x + 4$ et $y = 3x^2 + 4$.
Il y a aussi un lien à faire entre la sortie qui utilise la fonction `print()` et l'image des deux équations.

? Quelle stratégie veux-tu essayer afin de résoudre le problème?

J'utilise Python pour programmer ce problème. Mon programme doit :

- utiliser des commentaires afin d'expliquer les procédures du code;
- utiliser la boucle bornée pour l'intervalle de x entre -2 et 2 inclusivement;
- utiliser la fonction `print()` afin d'afficher mes résultats;
- exécuter le programme afin de faire ressortir l'information qu'il contient.

a) pour $y = 3x + 4$



STRATÉGIE 1

En utilisant un intervalle de données et une boucle bornée

```

1  # Programme qui affiche une table de valeurs pour l'équation  $y = 3x + 4$  pour l'intervalle
   -2 ≤ x ≤ 2
2
3  # Afficher l'entête de la table de valeurs
4  print(" x   ,   y")
5  print("-----")
6
7  # Établissement de l'intervalle de données à la variable x.
8  for x in [-2,-1,0,1,2] :
9      # J'affiche le tableau des valeurs pour l'équation  $y = 3x + 4$ 
10     print(x, "\t, ", 3*x+4)

```

```

x   ,   y
-----
-2   , -2
-1   ,  1
 0   ,  4
 1   ,  7
 2   , 10

```

Ce programme peut être consulté [en ligne](#).



STRATÉGIE 2

En utilisant la fonction `range()` pour créer l'intervalle de données et une boucle bornée

```
1 # Programme qui affiche une table de valeurs pour l'équation y = 3x + 4 pour l'intervalle
  -2 ≤ x ≤ 2
2
3 # Afficher l'entête de la table de valeurs
4 print(" x , y")
5 print("-----")
6
7 # Établissement de l'intervalle de données à la variable x en utilisant range().
8 for x in range(-2,3) :
9     # J'affiche le tableau des valeurs pour l'équation y = 3x + 4
10    print(x, "\t", 3*x+4)
```

```
x , y
-----
-2 , -2
-1 , 1
0 , 4
1 , 7
2 , 10
```

Ce programme peut être consulté [en ligne](#).



Pourquoi dois-tu inclure le 3 dans la fonction `range()` ?

b) $y = 3x^2 + 4$

En utilisant un intervalle de données avec une boucle non bornée

```
1 # Programme qui affiche une table de valeurs pour l'équation y =
  3x^2 + 4 pour l'intervalle -2 ≤ x ≤ 2.
2
3 # Afficher l'entête de la table de valeurs.
4 print(" x , y")
5 print("-----")
6
7 # Établissement de la valeur initiale de x.
8 x = -2
9 while x <= 2:
10    # Afficher le tableau des valeurs pour l'équation y = 3x^2 + 4.
11    print(x, "\t", 3 * x**2 + 4)
12    x = x + 1 # Augmenter la valeur de x de 1.
13
```

```
x , y
-----
-2 , 16
-1 , 7
0 , 4
1 , 7
2 , 16
```

Ce programme peut être consulté [en ligne](#).

Retour sur l'exemple 1

- Demander à quelques élèves de faire part au groupe-classe de leur solution et d'expliquer les stratégies utilisées pour créer une table de valeurs. Inviter les autres élèves à poser des questions afin de vérifier leur compréhension.
- Pendant les discussions, s'assurer que les élèves établissent des liens entre la production de tables de valeurs et le codage et que les éléments de la section **Éléments à faire ressortir** sont abordés. Si nécessaire, poser une question telle que : Quelle est la meilleure façon de représenter les données dans une table de valeurs en programmation?
- Encourager les élèves à ajouter des éléments à leur travail ou à consigner les éléments essentiels dans leur cahier de notes.

Déroulement pour l'exemple 2

- Présenter aux élèves l'exemple 2, soit Une relation linéaire ou non linéaire.
- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre diverses stratégies pour représenter des données et créer un graphique.
- Consulter les notes pédagogiques fournies dans le corrigé de l'exemple 2 pour obtenir des stratégies d'accompagnement au niveau des processus mathématiques et des éléments de difficultés que les élèves risquent de rencontrer.
- Choisir différentes stratégies de regroupements flexibles afin de répondre aux différents besoins des élèves.

Stratégie de résolution de problèmes pour l'exemple 2

L'élève utilise un logiciel de programmation ainsi que de la bibliothèque de création graphique Matplotlib afin de vérifier si une relation est linéaire ou non linéaire.

EXEMPLE 2

Une relation linéaire ou non linéaire

Pour chacun des exemples ci-dessous, à l'aide d'un outil de programmation et d'un graphique, vérifie si la relation est linéaire ou non linéaire.

a)

x	y
-4	14
-2	10
-1	8
3	0
8	-10

b)

x	y
-4	42
-2	14
1	8
3	24
6	70

Corrigé

Notes pédagogiques

- L'élève utilise ses connaissances antérieures en algèbre en faisant ressortir le taux de variation lors de la résolution du problème. L'élève compare ses taux de variation à l'aide d'un outil de programmation.
- L'élève ne doit pas nécessairement afficher le taux de variation pour chacun des x et des y . Ceux-ci peuvent cependant lui être utiles afin de confirmer ou vérifier la vraisemblance de la réponse.

J'utilise Python pour programmer ce problème. Mon programme doit :

- utiliser des commentaires afin d'expliquer les procédures du code;
- associer une variable à chacune des valeurs de y du tableau des valeurs;
- écrire une équation qui permet de déterminer les taux de variation;
- utiliser la fonction `print()` afin d'afficher mes résultats pour les taux de variation;
- exécuter le programme afin de faire ressortir l'information qu'il contient.

Notes pédagogiques

- L'élève utilise ses connaissances antérieures en algèbre en faisant ressortir le taux de variation lors de la résolution du problème. L'élève compare ses taux de variation à l'aide d'un outil de programmation.
- L'élève ne doit pas nécessairement afficher le taux de variation pour chacun des x et y . Ceux-ci peuvent cependant lui être utiles afin de confirmer ou vérifier la vraisemblance de la réponse.

a) Je détermine si la relation entre x et y est linéaire ou non linéaire.



STRATÉGIE 1

En utilisant le taux de variation

Je détermine si le taux de variation est constant.

	x	y	
tv_{xa}	$xa = -4$	$ya = 14$	tv_{ya}
tv_{xb}	$xb = -2$	$yb = 10$	tv_{yb}
tv_{xc}	$xc = -1$	$yc = 8$	tv_{yc}
tv_{xd}	$xd = 3$	$yd = 0$	tv_{yd}
	$xe = 8$	$ye = -10$	


```

1 # Programme qui trouve le taux de variation dans une table de valeurs
2
3 # Assigner les valeurs de la table à des variables
4 xa = -4
5 xb = -2
6 xc = -1
7 xd = 3
8 xe = 8
9 ya = 14
10 yb = 10
11 yc = 8
12 yd = 0
13 ye = -10
14
15 # Déterminer les taux de variation.
16 TVA = (yb-ya)/(xb-xa)
17 TVB = (yc-yb)/(xc-xb)
18 TVC = (yd-yc)/(xd-xc)
19 TVD = (ye-yd)/(xe-xd)
20
21 # Afficher les taux de variation.
22 print ("TVA =", TVA)
23 print ("TVB =", TVB)
24 print ("TVC =", TVC)
25 print ("TVD =", TVD)
26 # Vérifier si les taux de variation sont constants ou non constants.
27 ▼ if (TVA == TVB) and (TVB == TVC) and (TVC == TVD):
28     print("La relation est linéaire.")
29 ▼ else:
30     print("La relation est non linéaire.")

```

```

TVA = -2.0
TVB = -2.0
TVC = -2.0
TVD = -2.0
La relation est linéaire.

```

Ce programme peut être consulté [en ligne](#).

À l'aide de l'outil pédagogique, je détermine que le taux de variation est constant, je conclus donc que la relation est **linéaire**.

Notes pédagogiques

- L'élève utilise ses connaissances antérieures en algèbre en traçant un graphique à l'aide de la table de valeurs proposée. L'élève analyse le graphique afin de répondre à la question du problème.



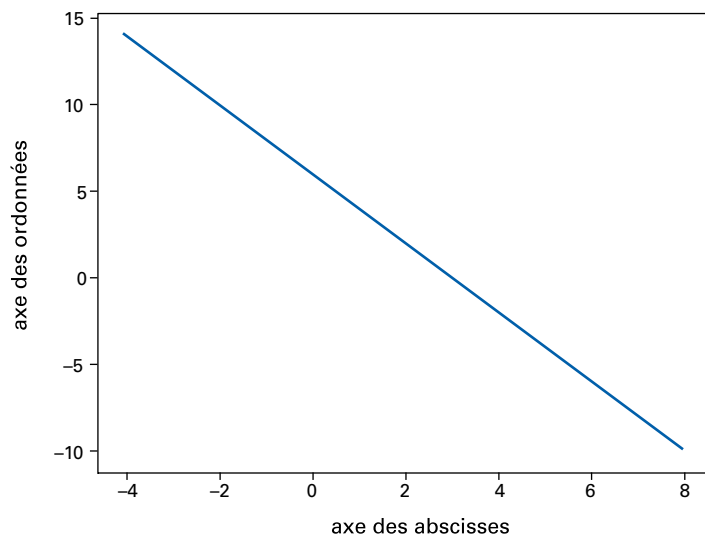
STRATÉGIE 2

En utilisant une représentation graphique



Est-ce que l'utilisation d'un graphique avec Python offre une meilleure analyse?

```
1 # Programme qui utilise la bibliothèque graphique de Python matplotlib
2 # pour créer un graphique afin de déterminer si une relation est linéaire ou non.
3
4 # Importer l'objet graphique pyplot et affectation à la variable plt.
5 import matplotlib.pyplot as plt
6
7 # Établissement de deux listes pour créer le graphique
8 x = [-4, -2, -1, 3, 8]
9 y = [14, 10, 8, 0, -10]
10
11 # Nommer les axes
12 plt.xlabel("axe des abscisses")
13 plt.ylabel("axe des ordonnées")
14
15 # Générer le graphique
16 plt.plot(x, y)
17 plt.show()
```



Ce programme peut être consulté [en ligne](#).

À l'aide de l'outil de programmation et du graphique, je constate que la relation est **linéaire**.

b) Je détermine si la relation entre x et y est linéaire ou non linéaire.

Notes pédagogiques

- L'élève utilise ses connaissances antérieures en algèbre en traçant un graphique à l'aide de la table de valeurs. L'élève analyse le graphique afin de répondre à la question du problème.
- L'élève ayant des connaissances avancées préalables en programmation pourrait vouloir utiliser un langage de programmation différent et même différentes fonctions plus avancées. L'enseignante ou l'enseignant devrait encourager les élèves à utiliser des fonctions plus avancées en programmation. L'élève pourrait utiliser une fonction permettant la prise de mémoire dans le code.
- Afin d'amener l'élève à déterminer la relation entre les variables x et y , il est important de fournir des pistes de réflexion ou un questionnement adéquat tout au long de la résolution du problème. Voici quelques exemples de questions ou de pistes de réflexion afin d'aider l'élève à simplifier des expressions algébriques.
 - En programmation, est-il plus avantageux d'utiliser la table de valeurs ou le graphique pour déterminer si une relation est linéaire ou non?
 - Quelles fonctions pourraient être ajoutées afin de rendre le graphique plus complet?



STRATÉGIE 1

En utilisant le taux de variation

Je détermine si le taux de variation est constant selon l'augmentation de x et de y .

	x	y	
tv_{xa}	$xa = -4$	$ya = 42$	tv_{ya}
tv_{xb}	$xb = -2$	$yb = 14$	tv_{yb}
tv_{xc}	$xc = -1$	$yc = 8$	tv_{yc}
tv_{xd}	$xd = 3$	$yd = 24$	tv_{yd}
	$xe = 6$	$ye = 70$	

```

1  # Programme qui trouve le taux de variation dans une table de valeurs
2
3  # Assigner les valeurs de la table à des variables
4  xa = -4
5  xb = -2
6  xc = -1
7  xd = 3
8  xe = 6
9  ya = 42
10 yb = 14
11 yc = 8
12 yd = 24
13 ye = 70
14
15 # Déterminer les taux de variation.
16 TVA = (yb-ya)/(xb-xa)
17 TVB = (yc-yb)/(xc-xb)
18 TVC = (yd-yc)/(xd-xc)
19 TVD = (ye-yd)/(xe-xd)
20
21 # Afficher les taux de variation.
22 print ("TVA =", TVA)
23 print ("TVB =", TVB)
24 print ("TVC =", TVC)
25 print ("TVD =", TVD)
26 # Vérifier si les taux de variation sont constants ou non constants.
27 ▼ if (TVA == TVB) and (TVB == TVC) and (TVC == TVD):
28     print("La relation est linéaire.")
29 ▼ else:
30     print("La relation est non linéaire.")

```

```

TVA = -14.0
TVB = -6.0
TVC = 4.0
TVD = 15.333333333333334
La relation est non linéaire.
>

```

Ce programme peut être consulté [en ligne](#).

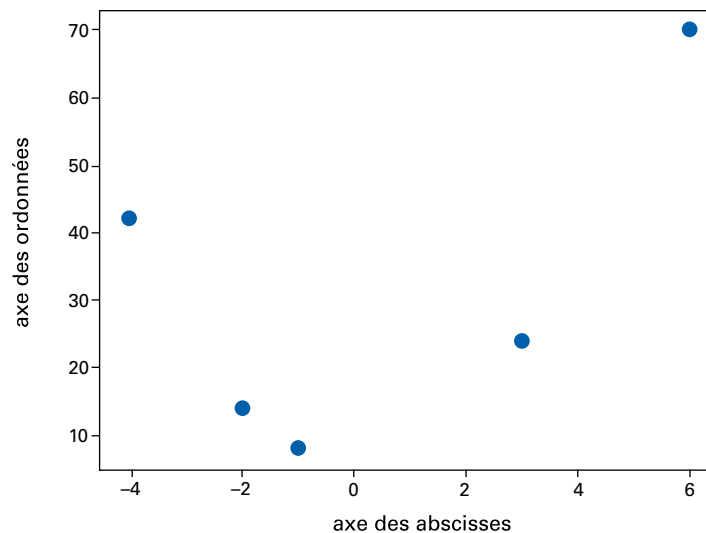
À l'aide de l'outil pédagogique, je détermine que le taux de variation est non constant, je conclus donc que la relation est **non linéaire**.



STRATÉGIE 2

En utilisant une représentation graphique

```
1 # Programme qui utilise la bibliothèque graphique de Python matplotlib
2 # pour créer un graphique afin de déterminer si une relation est linéaire ou non.
3
4 # Importer l'objet graphique pyplot et affectation à la variable plt.
5 import matplotlib.pyplot as plt
6
7 # Établissement de deux listes pour créer le graphique
8 x = [-4, -2, -1, 3, 6]
9 y = [42, 14, 8, 24, 70]
10
11 # Nommer les axes
12 plt.xlabel("axe des abscisses")
13 plt.ylabel("axe des ordonnées")
14
15 # Générer le graphique
16 plt.scatter(x, y)
17 plt.show()
```



Ce programme peut être consulté [en ligne](#).

À l'aide de l'outil de programmation et du graphique, je constate que la relation est **non linéaire**.

Retour sur l'exemple 2

- Demander à quelques élèves de faire part au groupe-classe de leur solution et d'expliquer les stratégies utilisées pour faire ressortir la relation linéaire et non linéaire dans le code à l'aide d'une condition. Inviter les autres élèves à poser des questions afin de vérifier leur compréhension.
- Pendant les discussions, s'assurer que les élèves établissent des liens entre l'utilisation des conditions et des boucles et que les éléments de la section **Éléments à faire ressortir** sont abordés. Si nécessaire, poser des questions telles que :
 - Comment peux-tu créer un graphique en programmation Python?
 - À quoi servent les conditions? Quelle est la différence entre les divers types de conditions?
- Encourager les élèves à ajouter des éléments à leur travail ou à consigner les éléments essentiels dans leur cahier de notes.

Déroulement pour l'exemple 3

- Présenter aux élèves l'exemple 3, soit Les formes géométriques.
- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre diverses stratégies pour reproduire des formes géométriques simples et utilisant des caractères comme un astérisque « * ».
- Consulter les notes pédagogiques fournies dans le corrigé de l'exemple 3 pour obtenir des stratégies d'accompagnement au niveau des processus mathématiques et des éléments de difficultés que les élèves risquent de rencontrer.
- Choisir différentes stratégies de regroupements flexibles afin de répondre aux différents besoins des élèves.

Stratégie de résolution de problèmes pour l'exemple 3

L'élève utilise un logiciel de programmation et utilise des boucles bornées et non bornées afin de reproduire des formes géométriques.

Notes pédagogiques

- Cet exemple permet à l'élève d'appliquer le processus mathématique de sélection d'outils et de stratégies.
- Il y a plusieurs stratégies possibles pour reproduire les formes géométriques ci-dessous en utilisant un outil de programmation. L'élève doit déterminer si une ou plusieurs boucles bornées ou non bornées doivent être utilisées.

EXEMPLE 3

Les formes géométriques

En utilisant des boucles, reproduis les formes géométriques suivantes.

a) * * * * *
* * * * *
* * * * *
* * * * *
* * * * *

b) *
* *
* * *
* * * *
* * * * *

Corrigé

Notes pédagogiques

- L'élève utilise ses connaissances antérieures en géométrie et mesure en faisant ressortir les propriétés des régularités pour reproduire les formes géométriques.

J'utilise Python pour programmer ce problème. Mon programme doit :

- utiliser des commentaires afin d'expliquer les procédures du code;
- associer une variable à chacune des valeurs de y dans le tableau des valeurs;
- écrire une équation qui permet de déterminer les taux de variation;
- utiliser la fonction `print()` afin d'afficher mes résultats pour les taux de variation;
- exécuter le programme afin de faire ressortir l'information qu'il contient.

a) J'utilise un outil de programmation pour créer la forme géométrique.

Notes pédagogiques

- L'élève utilise des boucles bornées imbriquées pour reproduire le carré de taille 5. Il est possible que l'élève utilise 5 boucles séparées, soit une pour chaque rangée. Poser les questions suivantes afin d'amener l'élève à améliorer son code :
 - Comment est-il possible de regrouper les boucles afin de simplifier le code?
 - Quelle est la relation entre le nombre de rangées et le nombre d'étoiles dans la forme?
- La fonction `print()` en Python saute automatiquement une ligne à chaque fois si une commande de fin de ligne n'est pas incluse. Par conséquent, il faut utiliser un deuxième paramètre pour garder le curseur sur la même ligne. La commande suivante affiche une étoile avec une espace sans sauter de ligne.

```
print(" * ", end = " ")
```

Je remarque qu'il y a autant d'étoiles sur l'horizontal que le vertical. J'utilise donc une boucle qui imprime chaque étoile sur l'horizontale (5 fois) et la place dans une boucle qui répète cette opération sur la verticale (5 fois).



STRATÉGIE 1

En utilisant des boucles bornées imbriquées

```
1 # Programme qui crée une forme en utilisant des boucles bornées
2
3 Forme = "*" # Utiliser une étoile pour créer la forme
4
5 # Boucle extérieure pour le nombre de rangées
6 ▼ for y in range(5):
7     # Boucle imbriquée pour le nombre d'étoiles
8     ▼ for x in range(5):
9         print(Forme, end = " ") # La commande "end" empêche le
    saut de ligne après la fonction print()
10     print("") # Sauter une ligne.
11
```

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Ce programme peut être consulté [en ligne](#).



STRATÉGIE 2

En utilisant des boucles non bornées imbriquées

Notes pédagogiques

L'élève peut également utiliser des boucles non bornées pour reproduire l'image. En utilisant une variable pour la taille de la forme géométrique (Taille = 5), le code devient plus versatile et peut afficher des formes géométriques de différentes tailles.

```
1 # Programme qui crée une forme en utilisant des boucles non
  bornées
2
3 Forme = "*" # Utiliser une étoile pour créer la forme
4 Taille = 5
5 # Boucle extérieure pour le nombre de rangées
6 y = 1
7 while y <= Taille:
8     y = y + 1
9     x = 1 # réinitialiser la valeur de x
10    # Boucle imbriquée pour le nombre d'étoiles
11    while x <= Taille:
12        x = x + 1 # augmenter la valeur de x
13        print(Forme, end = " ") # La commande "end" empêche le saut
    de ligne après la fonction print()
14    print("\n") # Sauter une ligne.
15
```

```
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
```

Ce programme peut être consulté [en ligne](#).

b) J'utilise un outil de programmation pour créer la forme géométrique.

Je remarque qu'il y a 5 rangées et que chaque rangée a un nombre croissant d'étoiles.



STRATÉGIE 1

En utilisant des boucles non bornées imbriquées

Notes pédagogiques

- Il est possible que l'élève éprouve certaines difficultés à reproduire l'image du triangle, car chaque rangée contient un nombre différent d'étoiles. Poser les questions suivantes pour appuyer l'élève dans sa réflexion :
 - Quelle est la relation entre le nombre d'étoiles et la rangée dans laquelle elles se trouvent?

- Combien de boucles sont nécessaires pour reproduire l'image du triangle?
- Quelle relation existe-t-il entre la largeur et la hauteur de la forme géométrique?

Je remarque que chaque rangée (rangée 1, rangée 2, etc.) contient le même nombre d'étoiles que son numéro, c'est-à-dire, rangée 1 = 1 étoile, rangée 2 = 2 étoiles, etc.

Je crée deux boucles non bornées et imbriquées qui impriment le nombre d'étoiles selon la rangée.

```

1  # Programme qui crée une forme en utilisant des boucles non
   bornées.
2
3  Forme = "*" # Utiliser une étoile pour créer la forme
4  Rangees = 5
5
6  # Boucle extérieure pour le nombre de rangées
7  y = 1
8  while y <= Rangees:
9      x = 1 # réinitialiser la valeur de x
10     # Boucle imbriquée pour le nombre d'étoiles
11     while x <= y:
12         x += 1 # Une autre façon d'augmenter la valeur de x par 1
13         print(Forme, end = " ") # La commande "end" empêche le saut
   de ligne après la fonction print()
14     y = y + 1 # Augmenter la valeur de y
15     print("") # Sauter une ligne.
16

```

```

*
* *
* * *
* * * *
* * * * *

```

Ce programme peut être consulté [en ligne](#).



STRATÉGIE 2

En utilisant une boucle bornée et une suite

Notes pédagogiques

- Cette stratégie permet d'utiliser une suite pour reproduire chaque rangée du triangle. Le nombre d'étoiles qui se trouve sur chaque rangée est indiqué dans la liste « Suite [] ». La stratégie utilise également un opérateur de multiplication avec la variable « *Forme* » pour reproduire le nombre d'étoiles par rangée.
- Il est possible que l'élève ne pense pas à cette stratégie, mais elle ou il doit être encouragé à trouver et faire le lien entre la suite et les nombres d'étoiles.

Je remarque que chaque rangée de la forme représente une suite linéaire croissante par 1. J'utilise donc une boucle et les valeurs dans une liste pour créer la forme.

```
1 # Programme qui crée une forme en utilisant une suite et une boucle bornées.
2
3 Forme = "*" # Utiliser une étoile pour créer la forme
4 Suite = [1,2,3,4,5]
5
6 # Boucle pour le nombre de rangées
7 for y in Suite:
8     print(Forme*y) # Imprimer la Forme, y fois.
```

```
*
* *
* * *
* * * *
* * * * *
```

Ce programme peut être consulté [en ligne](#).

Retour sur l'exemple 3

- Demander à quelques élèves de faire part au groupe-classe de leur solution et d'expliquer les stratégies utilisées pour reproduire les formes géométriques avec du code. Inviter les autres élèves à poser des questions afin de vérifier leur compréhension.
- Pendant les discussions, s'assurer que les élèves établissent des liens entre la géométrie et la programmation et que les éléments de la section **Éléments à faire ressortir** sont abordés. Si nécessaire, poser des questions telles que :
 - Comment l'analyse d'une forme géométrique peut-elle t'aider à créer du code pour la reproduire?
 - Est-ce que toutes les formes géométriques ont des régularités qui se prêtent au codage?
- Encourager les élèves à ajouter des éléments à leur travail ou à consigner les éléments essentiels dans leur cahier de notes.

ALLER PLUS LOIN

1. À l'aide d'un outil de programmation, démontre dans un graphique la valeur d'un placement garanti de 2 000 \$ à un taux de 3,0 % d'intérêt composé annuellement après 15 années d'épargne.
2. À l'aide d'un outil de programmation, reproduis la forme géométrique suivante :

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * *  
* * * *  
* * *  
* *  
*
```

Partie 2 – Apprentissage autonome

Déroulement

- Au besoin, demander aux élèves de faire quelques exercices de la section **À ton tour!**. Ces exercices peuvent servir de billet de sortie ou autres preuves d'apprentissage.
- Recueillir les preuves d'apprentissage des élèves et les interpréter selon l'atteinte des critères d'évaluation et cibler les prochaines étapes de l'enseignement et des élèves en vue de les aider à s'améliorer.

Note : Consulter le corrigé de la partie 2, s'il y a lieu.

Corrigé

1. À l'aide d'un outil de programmation, détermine si les valeurs de a sont inférieures, supérieures ou égales aux valeurs de b correspondantes.

a	b
4^{-2}	$\frac{1}{8}$
0,003	3×10^{-3}
$43\left(\frac{1}{4}\right)$	$43\left(\frac{1}{16}\right)$
5	$\left(\frac{1}{5}\right)^2$

Tu peux utiliser la fonction Python `zip(liste1, liste2)` qui permet de passer au travers de deux listes (suites) en même temps dans une boucle comme suit :

```
for a,b in zip(liste1, liste2) :  
    instructions  
    ...
```



Processus mathématique préconisé : établissement de liens

L'élève établit des liens entre les mathématiques et leur utilisation dans le domaine de la programmation. L'enseignante ou l'enseignant peut poser les questions suivantes afin d'aider l'élève à résoudre le problème :

- As-tu déjà utilisé une stratégie semblable? A-t-elle été efficace?
- Quels liens fais-tu avec ce que tu connais déjà en mathématiques?

J'utilise Python pour programmer ce problème. Mon programme doit :

- utiliser des commentaires afin d'expliquer les procédures du code;
- donner une valeur aux variables x et y ;
- utiliser les conditions et les opérateurs de comparaison afin de déterminer la relation entre x et y ;
- utiliser la fonction `print()` afin d'afficher mes résultats;
- exécuter le programme afin de faire ressortir l'information qu'il contient.



STRATÉGIE 1

En utilisant 4 structures conditionnelles distinctes

```
1 # Programme qui compare deux valeurs à l'aide d'une structure conditionnelle
  if..elif..else:
2
3 # Comparaison 1
4 a = 4**(-2)
5 b = 1/8
6 √ if a < b:
7     print(a,"est plus petit que",b)
8 √ elif a > b:
9     print(a,"est plus grand que",b)
10 √ else:
11     print(a,"est égale à",b)
12
13 # Comparaison 2
14 a = 0.003
15 b = 3E-3
16 √ if a < b:
17     print(a,"est plus petit que",b)
18 √ elif a > b:
19     print(a,"est plus grand que",b)
20 √ else:
21     print(a,"est égale à",b)
22
23 # Comparaison 3
24 a = 43*(1/4)
25 b = 43*(1/16)
26 √ if a < b:
27     print(a,"est plus petit que",b)
28 √ elif a > b:
29     print(a,"est plus grand que",b)
30 √ else:
31     print(a,"est égale à",b)
32
33 # Comparaison 4
34 a = 5
35 b = (1/5)**2
36 √ if a < b:
37     print(a,"est plus petit que",b)
38 √ elif a > b:
39     print(a,"est plus grand que",b)
40 √ else:
41     print(a,"est égale à",b)
```

```
0.0625 est plus petit que 0.125
0.003 est égale à 0.003
10.75 est plus grand que 2.6875
5 est plus grand que 0.040000000000000001
```

Ce programme peut être consulté [en ligne](#).



STRATÉGIE 2

En utilisant une boucle pour comparer deux intervalles de données

```
1 # Programme qui compare deux valeurs à l'aide d'une structure conditionnelle
  if..elif.,else:
2
3 # Établissement des deux listes pour les valeurs de a et b.
4 Liste_a = [4*(-2),0.003,43*(1/4),5]
5 Liste_b = [1/8,3E-3,43*(1/16),(1/5)**2]
6
7 # Établissement de la boucle
8 # zip() permet de passer au travers des deux listes en même temps
9 for a,b in zip(Liste_a, Liste_b):
10     if a < b:
11         print(a,"est plus petit que",b)
12     elif a > b:
13         print(a,"est plus grand que",b)
14     else:
15         print(a,"est égale à",b)
```

```
0.0625 est plus petit que 0.125
0.003 est égale à 0.003
10.75 est plus grand que 2.6875
5 est plus grand que 0.04000000000000001
```

Ce programme peut être consulté [en ligne](#).

2. À l'aide d'un outil de programmation, crée une table de valeurs pour l'intervalle $-10 \leq x \leq 10$ pour la relation $y = 5x - 6$.

Processus mathématique préconisé : sélection d'outils et de stratégies

L'élève sélectionne les outils et le format de la boucle bornée lui permettant de créer une table de valeurs pour l'intervalle d'une certaine relation. L'enseignante ou l'enseignant peut poser les questions suivantes afin d'aider l'élève à résoudre le problème :

- Quel outil serait utile pour résoudre le problème?
- Quelle serait la bonne stratégie pour résoudre ce problème?
- Quelle est la différence entre l'utilisation d'un intervalle de données ou l'utilisation de la fonction range?

J'utilise Python pour programmer ce problème. Mon programme doit :

- utiliser des commentaires afin d'expliquer les procédures du code;
- utiliser la boucle bornée pour l'intervalle de x entre 1 et 4 inclusivement;
- utiliser la fonction print() afin d'afficher mes résultats;
- exécuter le programme afin de faire ressortir l'information qu'il contient.



STRATÉGIE 1

En utilisant un intervalle de données pour $y = 5x - 6$

```
1 # Programme qui crée une table de valeurs à partir d'une liste.
2
3 # Afficher l'entête de la table de valeurs
4 print(" x , y")
5 print("-----")
6
7 # La boucle utilise une liste directement sans l'affecter à une variable.
8 ▼ for x in [-10,-9,-8,-7,-6,-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10] :
9   # Afficher les valeurs de x et y pour l'équation y = 5x - 6.
10  print(x, " , 5*x-6)
```

```
x , y
-----
-10 , -56
-9 , -51
-8 , -46
-7 , -41
-6 , -36
-5 , -31
-4 , -26
-3 , -21
-2 , -16
-1 , -11
0 , -6
1 , -1
2 , 4
3 , 9
4 , 14
5 , 19
6 , 24
7 , 29
8 , 34
9 , 39
10 , 44
▶
```

Ce programme peut être consulté [en ligne](#).



STRATÉGIE 2

En utilisant la fonction `range()` pour créer l'intervalle pour $y = 5x - 6$

```
1 # Programme qui crée une table de valeurs à partir d'une liste.
2
3 # Afficher l'entête de la table de valeurs
4 print(" x  ,  y")
5 print("-----")
6
7 # La boucle utilise la fonction range pour établir l'intervalle.
8 ▼ for x in range(-10, 11) :
9   # Afficher les valeurs de x et y pour l'équation y = 5x - 6.
10  print(x," ", 5*x-6)
```

```
x  ,  y
-----
-10 , -56
-9  , -51
-8  , -46
-7  , -41
-6  , -36
-5  , -31
-4  , -26
-3  , -21
-2  , -16
-1  , -11
0   , -6
1   , -1
2   , 4
3   , 9
4   , 14
5   , 19
6   , 24
7   , 29
8   , 34
9   , 39
10  , 44
```

Ce programme peut être consulté [en ligne](#).

3. À l'aide d'un outil de programmation et d'un graphique, vérifie si la relation est linéaire ou non linéaire.

x	y
-20	45
-8	8
-14	30
18	-20
3	-10

Processus mathématique préconisé : établissement de liens

L'élève établit des liens entre les mathématiques et leur utilisation dans le domaine de la programmation. L'enseignante ou l'enseignant peut poser les questions suivantes afin d'aider l'élève à résoudre le problème :

- As-tu déjà utilisé une stratégie semblable? A-t-elle été efficace?
- Quels liens fais-tu avec tes connaissances en algèbre?
- Quel est l'avantage d'utiliser la bibliothèque Matplotlib?



STRATÉGIE 1

En utilisant le taux de variation

Je détermine si le taux de variation est constant.

x	y
$xa = -20$	$ya = 45$
$xb = -8$	$yb = 8$
$xc = -14$	$yc = 30$
$xd = 18$	$yd = -20$
$xe = 3$	$ye = -10$

```

1 # Programme qui trouve le taux de variation dans une table de valeurs
2
3 # Assigner les valeurs de la table à des variables
4 xa = -20
5 xb = -8
6 xc = -14
7 xd = 18
8 xe = 3
9 ya = 45
10 yb = 8
11 yc = 30
12 yd = 20
13 ye = 10
14
15 # Déterminer les taux de variation.
16 TVA = (yb-ya)/(xb-xa)
17 TVB = (yc-yb)/(xc-xb)
18 TVC = (yd-yc)/(xd-xc)
19 TVD = (ye-yd)/(xe-xd)
20
21 # Afficher les taux de variation.
22 print ("TVA =", TVA)
23 print ("TVB =", TVB)
24 print ("TVC =", TVC)
25 print ("TVD =", TVD)
26 # Vérifier si les taux de variation sont constants ou non constants.
27 ▼ if (TVA == TVB) and (TVB == TVC) and (TVC == TVD):
28     print("La relation est linéaire.")
29 ▼ else:
30     print("La relation est non linéaire.")

```

```

TVA = -3.0833333333333335
TVB = -3.6666666666666665
TVC = -0.3125
TVD = 0.6666666666666666
La relation est non linéaire.

```

Ce programme peut être consulté [en ligne](#).

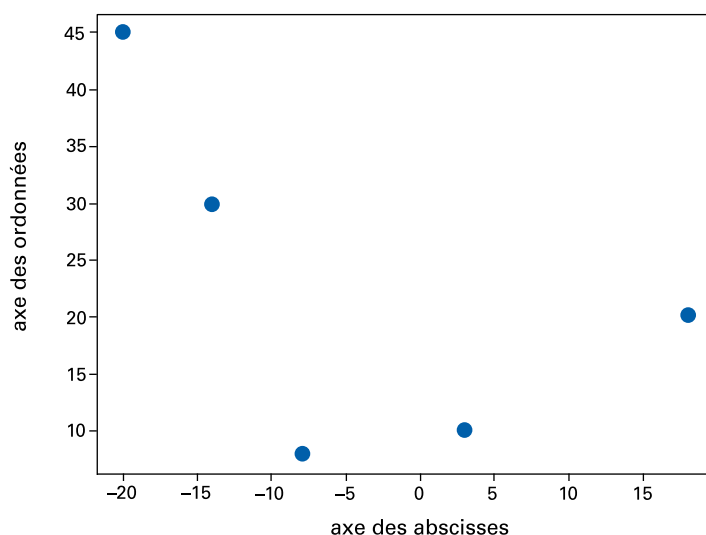
À l'aide de l'outil de programmation, je détermine que le taux de variation est non constant, donc je conclus que la relation est **non linéaire**.



STRATÉGIE 2

En utilisant une représentation graphique

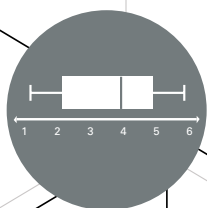
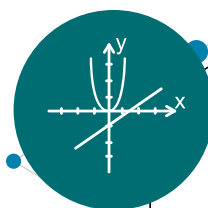
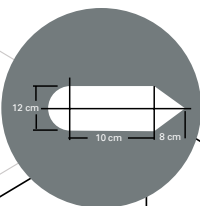
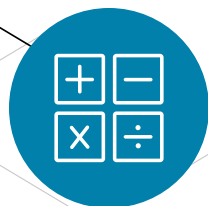
```
1 # Programme qui utilise la bibliothèque graphique de Python matplotlib
2 # pour créer un graphique afin de déterminer si une relation est linéaire ou non.
3
4 # Importer l'objet graphique pyplot et affecter à la variable plt.
5 import matplotlib.pyplot as plt
6
7 # Établissement de deux listes pour créer le graphique
8 x = [-20, -8, -14, 18, 3]
9 y = [45, 8, 30, 20, 10]
10
11 # Nommer les axes
12 plt.xlabel("axe des abscisses")
13 plt.ylabel("axe des ordonnées")
14
15 # Générer le graphique
16 plt.scatter(x, y)
17 plt.show()
```



Ce programme peut être consulté [en ligne](#).

À l'aide de l'outil de programmation et du graphique, je constate que la relation est **non linéaire**.

VERSION DE L'ÉLÈVE



$$\pi$$
$$1,4 \times 10^{-8}$$
$$-\frac{3}{4}$$

9^e
ANNÉE

PENSONS MATHÉMATIQUES!

Une approche renouvelée pour l'enseignement
et l'apprentissage des mathématiques

Minileçon

ALGÈBRE

Établir le lien entre les concepts
algébriques et le codage

Partie 1 – Exploration guidée

EXEMPLE 1

La création de tables de valeurs pour deux types de relation

À l'aide d'un outil de programmation, crée une table de valeurs pour l'intervalle $-2 \leq x \leq 2$ pour chacune des relations suivantes.

a) $y = 3x + 4$

b) $y = 3x^2 + 4$



TA STRATÉGIE

EXEMPLE 2

Une relation linéaire ou non linéaire

Pour chacun des exemples ci-dessous, à l'aide d'un outil de programmation et d'un graphique, vérifie si la relation est linéaire ou non linéaire.

a)

x	y
-4	14
-2	10
-1	8
3	0
8	-10

b)

x	y
-4	42
-2	14
1	8
3	24
6	70



TA STRATÉGIE

EXEMPLE 3

Les formes géométriques

En utilisant des boucles, reproduis les formes géométriques suivantes.

a) * * * * *
* * * * *
* * * * *
* * * * *
* * * * *

b) *
* *
* * *
* * * *
* * * * *



TA STRATÉGIE

Partie 2 – Apprentissage autonome

À ton tour!

1. À l'aide d'un outil de programmation, détermine si les valeurs de a sont inférieures, supérieures ou égales aux valeurs de b correspondantes.

a	b
4^{-2}	$\frac{1}{8}$
0,003	3×10^{-3}
$43\left(\frac{1}{4}\right)$	$43\left(\frac{1}{16}\right)$
5	$\left(\frac{1}{5}\right)^2$

Tu peux utiliser la fonction Python `zip(liste1, liste2)` qui permet de passer au travers de deux listes (suites) en même temps dans une boucle comme suit :

```
for a,b in zip(liste1, liste2) :  
    instructions  
    ...
```



TA STRATÉGIE

2. À l'aide d'un outil de programmation, crée une table de valeurs pour l'intervalle $-10 \leq x \leq 10$ pour la relation $y = 5x - 6$.



TA STRATÉGIE

3. À l'aide d'un outil de programmation et d'un graphique, vérifie si la relation est linéaire ou non linéaire.

x	y
-20	45
-8	8
-14	30
18	-20
3	-10



TA STRATÉGIE