

9^e
ANNÉE

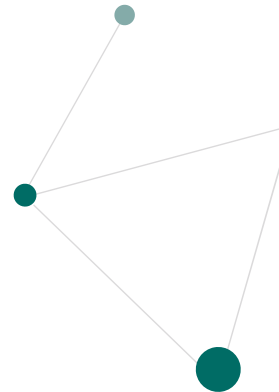
PENSONS MATHÉMATIQUES!

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

Minileçon

ALGÈBRE

Lire, prédire, analyser et modifier du code



RÉSUMÉ

Dans cette minileçon, l'élève lit du code afin de prédire son résultat. De plus, l'élève modifie du code, l'ajuste selon différentes contraintes et obtient des résultats.

PISTES D'OBSERVATION

L'élève :

- › interprète du code dans un langage de programmation;
- › prédit le résultat d'un code;
- › modifie un code afin de respecter les contraintes;
- › comprend et utilise des commandes permettant une saisie de données de la part de l'utilisateur.

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	Les expressions algébriques
Algèbre	La lecture, l'analyse et la modification du code
Algèbre	Les graphiques d'équations linéaires et non linéaires
Algèbre	Les graphiques d'inéquations linéaires et non linéaires

Partie 1 – Exploration guidée

Éléments à faire ressortir

- Un sous-programme permet de réutiliser des parties d'un code au lieu de les répéter ailleurs. En Python, un sous-programme s'appelle une fonction et est défini en utilisant la commande `def` suivie par deux points. Les instructions de la fonction sont indentées à la suite des deux points.

```
def {nom de la fonction}(paramètre1, paramètre2, ...) :  
    | instruction indentée  
    | instruction indentée
```

- La saisie d'information par l'utilisateur permet de créer des programmes plus généralisés. En Python, la saisie de valeurs se fait par l'entremise de la fonction `input()` :

```
s = input("Saisir la valeur texte ici : ")
```

Si la valeur est numérique, il faut ajouter la fonction `int()` à même celle de `input()` comme suit :

```
n = int(input("Saisir la valeur numérique ici : "))
```

- Une boucle est bornée lorsqu'il est possible de prédire exactement le nombre de fois que la boucle va se répéter.
- 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 et il est même possible qu'elle s'exécute à l'infini. Il n'est pas nécessaire de faire exécuter une boucle non bornée jusqu'à sa fin si une variable de contrôle est utilisée.
- Tous les langages de programmation fournissent une façon de produire un nombre aléatoire. Python fournit une bibliothèque de nombres qui génère des nombres aléatoires selon les restrictions imposées dans le code. Cette bibliothèque se nomme « `random` » et elle est importée via le code suivant :

```
import random
```
- Une structure conditionnelle (telle que `if...elif...else` : en Python) est utilisée lorsqu'il devient nécessaire d'exécuter une partie de code au lieu d'une autre.
- Un opérateur de comparaison est utilisé dans une structure conditionnelle et une boucle non bornée afin de comparer deux valeurs. Plusieurs comparaisons peuvent être faites en même temps en utilisant « `and` » ou « `or` ».

Déroulement pour l'exemple 1

- Consulter, au besoin, la fiche **La lecture, l'analyse et la modification du code** de la section **Concepts mathématiques** afin de revoir avec les élèves les stratégies de lecture et de compréhension d'un code de programmation ainsi que la terminologie liée à ces concepts en vue de les aider à réaliser l'activité. Dans cette minileçon, les élèves découvrent la fonction `input` et la bibliothèque `random`.
- Animer une discussion afin d'activer les connaissances antérieures ou connexes et d'établir des liens avec des expériences personnelles en posant des questions telles que :
 - Comment le code peut-il être plus interactif?
 - Quelle est l'importance de la comparaison dans du code?
- Présenter aux élèves l'exemple 1, soit L'analyse d'un code et la prédiction de sa sortie.
- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre diverses stratégies pour déterminer la sortie du code selon les éléments d'information dans celui-ci.
- 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 d'analyser un programme et de déterminer sa sortie.

Notes pédagogiques

Cet exemple permet à l'élève de mettre en application les processus mathématiques de communication et de réflexion.

EXEMPLE 1

L'analyse d'un code et la prédiction de sa sortie

Analyse le programme non documenté suivant et détermine son fonctionnement et un exemple de sortie.

```
1 import random
2
3 nb_essais = 0
4 nb_devine = 0
5
6 nom_joueur = input("Quel est ton nom? : ")
7
8 nombre = random.randint(1,50)
9
10 print("Salut ", nom_joueur, ", je pense à un entier entre 1 et 50. Peux-tu deviner ce
    nombre? Tu as 10 essais pour réaliser l'exercice.")
11
12 while nb_essais < 10 and nb_devine != nombre:
13     nb_devine = int(input("Nombre deviné : "))
14     nb_essais = nb_essais + 1
15     if nb_devine < nombre :
16         print("Ta valeur devinée est trop petite")
17     if nb_devine > nombre :
18         print("Ta valeur devinée est trop grande")
19
20 if nb_devine == nombre :
21     print("Bravo! Tu as deviné le bon nombre en ", str(nb_essais), "essais!")
22 else :
23     print("Tu n'as pas réussi à deviner. Je pensais au nombre ", str(nombre), ".")
24
```

Corrigé

Notes pédagogiques

- L'élève prend le temps de bien comprendre le programme et est amenée ou amené à réfléchir sur les différentes étapes ou stratégies à mettre en œuvre afin de résoudre le problème. L'élève utilise un vocabulaire précis afin de prédire la sortie du programme. L'élève est invitée ou invité à se soucier de la bonne utilisation des symboles et de la syntaxe lors de l'exécution d'un programme informatique.
- Ici, l'élève transcrit l'information véhiculée sur chacune des lignes de code afin de prédire la sortie du programme.

- Afin d'amener l'élève à communiquer sa solution, à faire une représentation efficace et à établir des liens entre l'entrée et la sortie d'un programme informatique, voici quelques exemples de questions ou de pistes de réflexion que l'enseignante ou l'enseignant pourrait utiliser :
 - Pourrais-tu écrire un code différent pour effectuer cette même tâche de programmation?
 - Y aurait-il une meilleure approche afin d'effectuer cette tâche?
 - Comment est-ce que ta solution se compare avec celles des autres élèves de ta classe?
- Demander à l'élève s'il y a une stratégie à prendre pour deviner le nombre choisi par le programme. Poser les questions suivantes pour l'aider dans sa réflexion :
 - Combien d'essais sont nécessaires pour trouver le nombre choisi par l'ordinateur?
 - Est-ce qu'il y a un avantage de savoir si le choix de l'utilisateur est trop bas ou trop élevé?

Je détermine la sortie du programme à l'aide de l'information présentée :

- (1) La première ligne définit l'importation d'une valeur aléatoire.
- (3-4) Définition des valeurs de départ. La première tient compte du nombre d'essais et la deuxième, du nombre deviné par l'utilisateur.
- (6) La ligne indique une saisie d'information par l'utilisateur. *Quel est ton nom?* :
- (8) La ligne permet au programme de choisir un nombre entier aléatoire de la console `import random`.
- (10) Le programme imprime une phrase qui indique les règlements du jeu.
- (12-18) Une boucle non bornée. Le programme demande à l'utilisateur de deviner une valeur numérique aléatoire, jusqu'à 10 essais. La boucle continue tant que l'utilisateur n'a pas fait 10 essais et que le nombre n'est pas deviné.
- (13-14) Saisie du nombre entier à deviner et augmentation du nombre d'essais.
- (15-16) Si la valeur choisie par l'utilisateur est plus petite que la valeur aléatoire, le message suivant apparaît : *Ta valeur devinée est trop petite.*
- (17-18) Si la valeur choisie par l'utilisateur est plus grande que la valeur aléatoire, le message suivant apparaît : *Ta valeur devinée est trop grande.*

Bibliothèque

`import random:`
Une bibliothèque de nombres qui génère des nombres pseudoaléatoires selon les restrictions imposées dans le code.

- (20-23) Affichage d'un message final selon la valeur saisie dans la boucle.
- (17-18) Si la valeur choisie par l'utilisateur est égale à la valeur aléatoire, le code sort de la boucle et actionne le code de la ligne 20 ou 22.
- (20-21) Si la valeur choisie par l'utilisateur est égale à la valeur aléatoire, le message suivant apparaît : *Bravo! Tu as deviné le bon nombre en nb_essais essais.*
- (22-23) Si la valeur choisie par l'utilisateur n'est jamais la bonne en 10 essais, le programme envoie le message suivant : *Mauvaise réponse. Je pensais au nombre « nombre ».*

Je détermine que ce code va créer un jeu-questionnaire qui permet à l'utilisateur de deviner un nombre aléatoire que le programme va sélectionner entre 1 et 50 inclusivement et lui donner 10 essais pour le trouver.

Voici un exemple de sortie du jeu :

```
Quel est ton nom? :
Salut , je pense à un entier entre 1 et 50. Peux-tu deviner ce nombre? Tu as 10 essais pour
réaliser l'exercice.
Nombre deviné : 25
Ta valeur devinée est trop grande
Nombre deviné : 10
Ta valeur devinée est trop petite
Nombre deviné : 15
Ta valeur devinée est trop petite
Nombre deviné : 18
Ta valeur devinée est trop petite
Nombre deviné : 22
Ta valeur devinée est trop petite
Nombre deviné : 24
Ta valeur devinée est trop grande
Nombre deviné : 23
Bravo! Tu as deviné le bon nombre en 7 essais!
> 
```

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 déterminer la sortie du programme. 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 les fonctions utilisées et la sortie du programme et que les éléments de la section **Éléments à faire ressortir** sont abordés. Si nécessaire, poser des questions telles que :
 - Pourquoi est-il nécessaire d'utiliser la bibliothèque random?
 - Pourrais-tu utiliser une boucle bornée au lieu d'une boucle non bornée pour effectuer la même tâche? Si oui, comment? Sinon, pourquoi pas?
 - Quelle est l'importance de l'utilisation des opérateurs de comparaison telle que « != » et « and »?
- 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 La modification d'un programme.
- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre diverses stratégies pour modifier un programme et faire respecter les nouveaux paramètres d'une mise en situation.
- 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 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 2

L'élève utilise un logiciel de programmation pour modifier un programme afin de respecter les nouveaux paramètres.

EXEMPLE 2

La modification d'un programme

Le programme ci-dessous permet de calculer l'aire d'un disque en demandant à l'utilisateur d'entrer le rayon.

```
1 ▼ def Aire_disque(Rayon):  
2     Pi = 3.14  
3     Aire_du_disque = # formule de l'aire du disque  
4     return Aire_du_disque  
5  
6     Rayon = int(input("Veuillez saisir le rayon du disque : "))
```

Modifie le programme selon les paramètres suivants :

- le programme doit déterminer la circonférence du cercle et l'aire du disque;
- le rayon doit être entré par l'utilisateur;
- le programme se termine lorsque la valeur 0 est entrée pour le rayon;
- la sortie du programme doit être la circonférence du cercle et l'aire du disque.

Utilise ton programme pour déterminer l'effet de la modification du rayon sur la circonférence et sur l'aire.

Corrigé

Notes pédagogiques

- Cet exemple permet à l'élève de mettre en application le processus mathématique de raisonnement et justification.
- L'élève prend le temps de bien comprendre le programme ainsi que la mise en situation proposée et est amené à réfléchir sur les différentes étapes ou stratégies à mettre en œuvre afin d'atteindre la sortie du programme demandée. L'élève doit utiliser une programmation efficace en se souciant de la bonne utilisation de la syntaxe et des symboles lors de l'exécution du programme informatique.

1. Je détermine les étapes pour la résolution du problème.
2. J'utilise un langage de programmation afin de déterminer la circonférence du cercle et l'aire du disque.
3. Je crée un pseudocode qui pourra être retranscrit en langage de programmation.

Notes pédagogiques

- Ici, l'élève pourrait avoir de la difficulté à comprendre la nomenclature derrière l'utilisation de la fonction `input`. Cette commande est nouvelle dans la série de commandes l'étude et il serait important de prendre le temps d'expliquer à l'élève son objectif et sa fonction explicitement.
- L'élève pourrait choisir d'utiliser un programme différent afin d'effectuer la même tâche. Il est de mise que l'enseignante ou l'enseignant encourage l'élève à utiliser ses connaissances antérieures afin de justifier les commandes utilisées lors de l'exécution du travail.

J'utilise un langage de programmation afin de modifier le programme.

Je modifie le sous-programme « `Aire_disque()` » afin d'ajouter l'équation pour l'aire d'un disque, soit $Aire_{\text{disque}} = \pi r^2$.

```
4 # Définition du sous-programme pour l'aire
5 def Aire_disque(Rayon):
6     Aire_du_disque = math.pi*(Rayon**2) # formule de l'aire du disque avec math.pi
7     return Aire_du_disque
```

J'ajoute un sous-programme « `Circonference_cercle()` » afin de pouvoir calculer la circonférence en utilisant la formule suivante : $Circonference_{\text{cercle}} = 2\pi r$

```
9 # Définition du sous-programme pour la circonférence
10 def Circonference_cercle(Rayon):
11     Pi = 3.14 # Une valeur est affectée à la variable Pi
12     Circonference_du_cercle = 2*Pi*Rayon
13     return Circonference_du_cercle
```

Je crée une boucle bornée qui saisit la valeur du rayon et qui affiche soit la circonférence du cercle et l'aire du disque, soit un message de fin de programme si la valeur saisie est 0 pour le rayon.

```
15 # Écriture du programme principal
16 Rayon = -1 # établissement d'une valeur initiale qui n'est pas 0.
17 while Rayon != 0:
18     Rayon = int(input("Veuillez saisir le rayon du cercle (en cm) : "))
19     if Rayon != 0:
20         print("La circonférence du cercle de rayon",Rayon,"est environ",
              round(Circonference_cercle(Rayon),1),"cm.")
21         print("L'aire du disque de rayon",Rayon,"est environ",round(Aire_disque(Rayon),1),"cm\u00b2.")
22         print("") # Sauter une ligne
23     else:
24         print("Fin du programme")
```

Dans les sous-programmes, j'utilise la constante `pi` provenant de la bibliothèque `math` pour remplacer la valeur de 3,14 dans le sous-programme car elle est plus précise.

Le programme final se trouve ci-dessous avec un exemple de sortie.

```
1 import math # Importer le module de math pour accéder la valeur de pi.
2 # Programme qui affiche la circonférence du cercle et l'aire du disque selon le rayon
  saisi.
3
4 # Définition du sous-programme pour l'aire
5 ▼ def Aire_disque(Rayon):
6     Aire_du_disque = math.pi*(Rayon**2) # formule de l'aire du disque avec math.pi
7     return Aire_du_disque
8
9 # Définition du sous-programme pour la circonférence
10 ▼ def Circonference_cercle(Rayon):
11     Pi = 3.14 # Une valeur est affectée à la variable Pi
12     Circonference_du_cercle = 2*Pi*Rayon
13     return Circonference_du_cercle
14
15 # Écriture du programme principal
16 Rayon = -1 # établissement d'une valeur initiale qui n'est pas 0.
17 ▼ while Rayon != 0:
18     Rayon = int(input("Veuillez saisir le rayon du cercle (en cm) : "))
19     ▼ if Rayon != 0:
20         print("La circonférence du cercle de rayon",Rayon,"est environ",
            round(Circonference_cercle(Rayon),1),"cm.")
21         print("L'aire du disque de rayon",Rayon,"est
            environ",round(Aire_disque(Rayon),1),"cm\u00b2.")
22         print("") # Sauter une ligne
23     ▼ else:
24         print("Fin du programme")
```

```
Veillez saisir le rayon du cercle (en cm) : 5
La circonférence du cercle de rayon 5 est environ 31.4 cm.
L'aire du disque de rayon 5 est environ 78.5 cm2.

Veillez saisir le rayon du cercle (en cm) : 10
La circonférence du cercle de rayon 10 est environ 62.8 cm.
L'aire du disque de rayon 10 est environ 314.2 cm2.

Veillez saisir le rayon du cercle (en cm) : 15
La circonférence du cercle de rayon 15 est environ 94.2 cm.
L'aire du disque de rayon 15 est environ 706.9 cm2.

Veillez saisir le rayon du cercle (en cm) : 20
La circonférence du cercle de rayon 20 est environ 125.6 cm.
L'aire du disque de rayon 20 est environ 1256.6 cm2.

Veillez saisir le rayon du cercle (en cm) : 0
Fin du programme
> |
```

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

Selon mes résultats, je constate que si le rayon d'un cercle est doublé,
la circonférence du cercle est doublée et l'aire du disque est 4 fois plus grande.

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 déterminer la circonférence du cercle et l'aire du disque. 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 :
 - Quelles sont les plus grandes difficultés pour modifier un code?
 - Comment ta modification du code se compare-t-elle avec celle de tes pairs?
- Encourager les élèves à ajouter des éléments à leur travail ou à consigner les éléments essentiels dans leur cahier de notes.

ALLER PLUS LOIN

À l'aide d'un logiciel de programmation, modifie le programme du périmètre du cercle et de l'aire du disque pour déterminer le périmètre et l'aire d'autres figures (par exemple, trapèze, rectangle, triangle).

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. Le programme suivant permet de calculer le taux de variation d'une droite passant par les points $(-3,0)$ et $(5,3)$. Modifie le code du programme afin de permettre à l'utilisateur de saisir 2 points sur une droite quelconque et d'afficher le taux de variation à l'aide d'un calcul dans une fonction.

```
1 # J'identifie les coordonnées du premier point
2 x1 = -3
3 y1 = 0
4 # J'identifie les coordonnées du deuxième point
5 x2 = 5
6 y2 = 3
7
8 Taux_variation = (y2-y1)/(x2-x1)
9 print (f"Le taux de variation de la droite passant par les points
    (-3,0) et (5,3) est égal à {Taux_variation}.")
```

Le taux de variation de la droite passant par les points $(-3,0)$ et $(5,3)$ est égal à 0.375.

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

Processus mathématique préconisé : établissement de liens

L'élève établit des liens entre les mathématiques et leur utilisation dans le domaine de la programmation.

L'élève établit aussi des liens entre les différentes fonctions possibles afin de résoudre le problème.

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 la fonction `input()` avec la fonction `int()` afin de saisir les valeurs des coordonnées de l'utilisateur.

J'insère le calcul du taux de variation à l'intérieur d'un sous-programme.

Je vérifie le bon fonctionnement de mon programme.

```
1▼ def tv(x1,y1,x2,y2):
2     return (y2-y1)/(x2-x1)
3
4 # J'identifie les coordonnées du premier point
5 x1 = int(input("Indiquer la valeur x du premier point : "))
6 y1 = int(input("Indiquer la valeur y du premier point : "))
7 # J'identifie les coordonnées du deuxième point
8 x2 = int(input("Indiquer la valeur x du deuxième point : "))
9 y2 = int(input("Indiquer la valeur y du deuxième point : "))
10
11 Taux_variation = tv(x1,y1,x2,y2)
12 print (f"Le taux de variation de la droite passant par les points ({x1},{y1}) et ({x2},
    {y2}) est égal à {Taux_variation}.")
```

```
Indiquer la valeur x du premier point : -3
Indiquer la valeur y du premier point : 0
Indiquer la valeur x du deuxième point : 5
Indiquer la valeur y du deuxième point : 3
Le taux de variation de la droite passant par les points (-3,0) et (5,3) est égal à 0.375.
>
```

Commande

f-string :

Une commande qui permet d'insérer des valeurs à des variables sans nécessairement ajuster le format de celles-ci. Cette commande est très utile lorsqu'on imprime un message provenant d'une commande `input`.

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

- 
2. Modifie le programme de l'exemple 1 afin de déterminer le taux de variation, l'abscisse et l'ordonnée à l'origine ainsi que l'équation de la droite lorsque 2 points de la droite sont entrés par l'utilisateur.

Processus mathématique préconisé : sélection d'outils et de stratégies

L'élève sélectionne les outils et les fonctions lui permettant de déterminer le pourcentage d'élèves qui utilisent l'autobus de ville. 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 stratégie serait bonne pour résoudre ce problème?
- Quel type de réponse est attendu : une réponse exacte ou une estimation?

Puisque le taux de variation est déjà trouvé dans l'exemple 1, soit $tv = \frac{y_2 - y_1}{x_2 - x_1}$, je peux utiliser l'équation générale d'une droite ($y = ax + b$) pour déterminer l'ordonnée à l'origine :

$$y = ax + b$$

$$y = (tv)x + b$$

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1} \right) x + b$$

$$y - \left(\frac{y_2 - y_1}{x_2 - x_1} \right) x = b$$

$$y - (tv)x = b$$

Je peux utiliser les coordonnées du premier ou du deuxième point pour remplacer les valeurs de x et y .

La valeur de y à l'abscisse à l'origine est toujours 0 donc, je peux déterminer l'équation de l'abscisse à l'origine en utilisant le taux de variation, l'ordonnée à l'origine et l'équation générale d'une droite :

$$y = ax + b$$

$$0 = (tv)x + b$$

$$-b = (tv)x$$

$$\frac{-b}{(tv)} = x$$

À l'aide de Python, je documente et produis le programme avec la sortie suivant.

```
1 # Programme qui détermine le taux de variation, l'ordonnée et l'abscisse à l'origine
  ainsi que l'équation de la droite à partir de deux points.
2
3 # Sous-programme pour le taux de variation
4 ▼ def tv(x1,y1,x2,y2):
5     return (y2-y1)/(x2-x1)
6
7 # Sous-programme pour l'origine à l'ordonnée.
8 ▼ def b(tv,x,y):
9     return y-tv*x
10
11 # J'identifie les coordonnées du premier point
12 x1 = int(input("Indiquer la valeur x du premier point : "))
13 y1 = int(input("Indiquer la valeur y du premier point : "))
14
15 # J'identifie les coordonnées du deuxième point
16 x2 = int(input("Indiquer la valeur x du deuxième point : "))
17 y2 = int(input("Indiquer la valeur x du deuxième point : "))
18
19 Taux_variation = tv(x1,y1,x2,y2)
20 # Utilisation de x1 et y1 pour trouver l'ordonnée à l'origine
21 Ordonnee_origine = b(Taux_variation,x1,y1)
22 print(f"\nLe taux de variation de la droite passant par les points ({x1},{y1}) et ({x2},
  {y2}) est égal à {Taux_variation}.")
23 print(f"L'ordonnée à l'origine de cette droite en utilisant le point ({x1},{y1}) est
  égale à {Ordonnee_origine}.")
24
25 # Utilisation de x2 et y2 pour trouver l'ordonnée à l'origine
26 Ordonnee_origine = b(Taux_variation,x2,y2)
27 print(f"Je vérifie que l'ordonnée à l'origine de cette droite en utilisant l'autre
  point ({x2},{y2}) est égale à {Ordonnee_origine}.")
28
29 # Calcul de l'abscisse à l'origine
30 Abscisse_origine = (-1)*Ordonnee_origine / Taux_variation
31 print(f"\nL'abscisse à l'origine de la droite passant par les points ({x1},{y1}) et
  ({x2},{y2}) est égale à {Abscisse_origine}.")
32
33 # Sortie de l'équation de la droite
34 ▼ if Ordonnee_origine >= 0:
35     print(f"L'équation de la droite passant par les points ({x1},{y1}) et ({x2},{y2}) est
      y = {Taux_variation}x + {Ordonnee_origine}")
36 ▼ else:
37     print(f"L'équation de la droite passant par les points ({x1},{y1}) et ({x2},{y2}) est
      y = {Taux_variation}x {Ordonnee_origine}")
```

```
Indiquer la valeur x du premier point : -3
Indiquer la valeur y du premier point : 0
Indiquer la valeur x du deuxième point : 5
Indiquer la valeur x du deuxième point : 3
```

```
Le taux de variation de la droite passant par les points (-3,0) et (5,3) est égal à 0.375.
L'ordonnée à l'origine de cette droite en utilisant le point (-3,0) est égale à 1.125.
Je vérifie que l'ordonnée à l'origine de cette droite en utilisant l'autre point (5,3)
est égale à 1.125.
```

```
L'abscisse à l'origine de la droite passant par les points (-3,0) et (5,3) est égale à -3.0.
L'équation de la droite passant par les points (-3,0) et (5,3) est y = 0.375x + 1.125
```

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

3. Analyse le programme non documenté suivant et détermine sa sortie.

```
1 import matplotlib.pyplot as plt
2
3 Valeurs_x = range(-100,100)
4 y1=[-4*x-6 for x in Valeurs_x]
5 y2=[-25 for x in Valeurs_x]
6
7 plt.xlabel("Axe des abscisses")
8 plt.ylabel("Axe des ordonnées")
9
10 plt.plot(Valeurs_x,y1)
11 plt.plot(Valeurs_x,y2)
12 plt.show()
```

Processus mathématique préconisé : communication

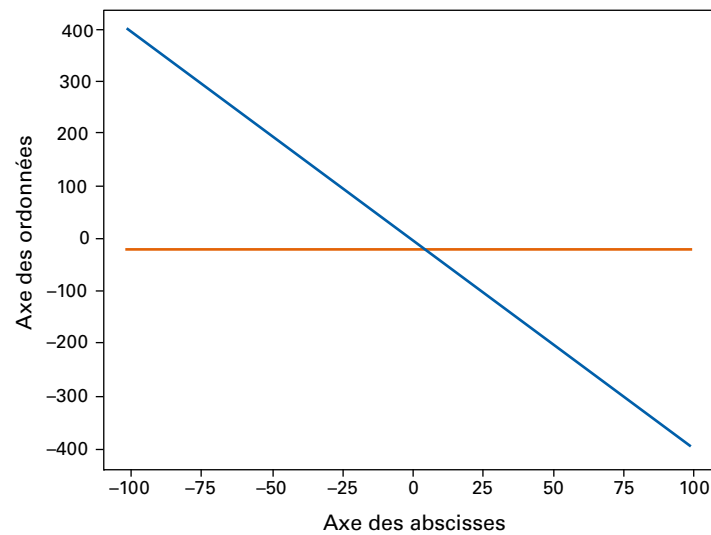
L'élève analyse et fait ressortir les éléments importants retrouvés dans le programme. L'enseignante ou l'enseignant peut poser les questions suivantes afin d'aider l'élève à résoudre le problème :

- Comment peux-tu expliquer ta démarche?
- Communique-tu clairement tes résultats et tes solutions?

Je détermine la sortie du programme à l'aide de l'information présentée :

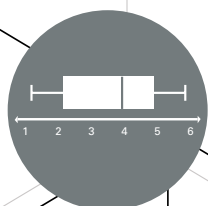
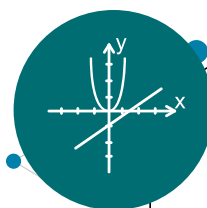
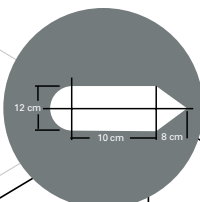
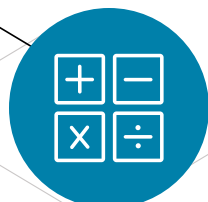
- (1) La première ligne définit l'importation d'un graphique à l'aide de Matplotlib.
- (3) La ligne indique l'intervalle des données pour la variable x , soit $-100 < x < 100$.
- (4-5) Les 2 lignes suivantes représentent les équations pour y , soit $y = -4x + 6$ et $y = -25$.
- (7) La ligne suivante définit le titre de l'axe des abscisses.
- (8) La ligne suivante définit le titre de l'axe des ordonnées.
- (9) Création d'un graphique à l'aide de la fonction `plt.plot (x, y)` pour l'équation $y = -4x + 6$.
- (10) Création d'un graphique à l'aide de la fonction `plt.plot (x, y)` pour l'équation $y = -25$.
- (9) Afficher le graphique avec `plt.show()`.

Je confirme le fonctionnement du programme à l'aide du graphique suivant.



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

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

Lire, prédire, analyser et modifier du code

Partie 1 – Exploration guidée

EXEMPLE 1

L'analyse d'un code et la prédiction de sa sortie

Analyse le programme non documenté suivant et détermine son fonctionnement et un exemple de sortie.

```
1  import random
2
3  nb_essais = 0
4  nb_devine = 0
5
6  nom_joueur = input("Quel est ton nom? : ")
7
8  nombre = random.randint(1,50)
9
10 print("Salut ", nom_joueur, ", je pense à un entier entre 1 et 50. Peux-tu deviner ce
    nombre? Tu as 10 essais pour réaliser l'exercice.")
11
12 while nb_essais < 10 and nb_devine != nombre:
13     nb_devine = int(input("Nombre deviné : "))
14     nb_essais = nb_essais + 1
15     if nb_devine < nombre :
16         print("Ta valeur devinée est trop petite")
17     if nb_devine > nombre :
18         print("Ta valeur devinée est trop grande")
19
20 if nb_devine == nombre :
21     print("Bravo! Tu as deviné le bon nombre en ", str(nb_essais), "essais!")
22 else :
23     print("Tu n'as pas réussi à deviner. Je pensais au nombre ", str(nombre), ".")
24
```



TA STRATÉGIE



EXEMPLE 2

La modification d'un programme

Le programme ci-dessous permet de calculer l'aire d'un disque en demandant à l'utilisateur d'entrer le rayon.

```
1 ▼ def Aire_disque(Rayon):  
2     Pi = 3.14  
3     Aire_du_disque = # formule de l'aire du disque  
4     return Aire_du_disque  
5  
6     Rayon = int(input("Veuillez saisir le rayon du disque : "))
```

Modifie le programme selon les paramètres suivants :

- le programme doit déterminer le périmètre du cercle et l'aire du disque;
- le rayon doit être entré par l'utilisateur;
- le programme se termine lorsque la valeur 0 est entrée pour le rayon;
- la sortie du programme doit être la circonférence du cercle et l'aire du disque.

Utilise ton programme pour déterminer l'effet de la modification du rayon sur la circonférence et sur l'aire.



TA STRATÉGIE

Partie 2 – Apprentissage autonome

À ton tour!

1. Le programme suivant permet de calculer le taux de variation d'une droite passant par les points $(-3,0)$ et $(5,3)$. Modifie le code du programme afin de permettre à l'utilisateur de saisir 2 points sur une droite quelconque et d'afficher le taux de variation à l'aide d'un calcul dans une fonction.

```
1 # J'identifie les coordonnées du premier point
2 x1 = -3
3 y1 = 0
4 # J'identifie les coordonnées du deuxième point
5 x2 = 5
6 y2 = 3
7
8 Taux_variation = (y2-y1)/(x2-x1)
9 print(f"Le taux de variation de la droite passant par les points
    (-3,0) et (5,3) est égal à {Taux_variation}.")
```

```
Le taux de variation de la droite passant par les points (-3,0) et (5,3) est égal à 0.375.
```



TA STRATÉGIE

2. Modifie le programme de l'exemple 1 afin de déterminer la variation, l'abscisse et l'ordonnée à l'origine ainsi que l'équation de la droite lorsque 2 points de la droite sont entrés par l'utilisateur.



TA STRATÉGIE

3. Analyse le programme non documenté suivant et détermine sa sortie.

```
1 import matplotlib.pyplot as plt
2
3 Valeurs_x = range(-100,100)
4 y1=[-4*x-6 for x in Valeurs_x]
5 y2=[-25 for x in Valeurs_x]
6
7 plt.xlabel("Axe des abscisses")
8 plt.ylabel("Axe des ordonnées")
9
10 plt.plot(Valeurs_x,y1)
11 plt.plot(Valeurs_x,y2)
12 plt.show()
```



TA STRATÉGIE