

9^e
ANNÉE

PENSONS MATHÉMATIQUES!

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

Minileçon

ALGÈBRE

Utiliser les concepts de base en programmation

RÉSUMÉ

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

PISTES D'OBSERVATION

L'élève :

- › utilise un logiciel de programmation pour évaluer des expressions algébriques et numériques simples;
- › utilise des opérateurs mathématiques lors de sa programmation;
- › utilise la syntaxe dans un langage de programmation;
- › utilise des variables afin d'écrire des programmes simples;

MATÉRIEL

- › ordinateur;
- › logiciel de programmation en ligne ou local.

Suggestion : replit.com fournit plusieurs langages de programmation pour subvenir aux besoins de cette minileçon.

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
Nombres	Le signe et la valeur des exposants
Nombres	Les opérations sur la notation scientifique
Nombres	Les opérations sur les puissances
Algèbre	Les concepts de base de programmation

Partie 1 – Exploration guidée

Éléments à faire ressortir

- Les commentaires sont très utiles pour comprendre la séquence d'un programme et doivent être utilisés le plus possible dans le codage.
- Les opérateurs sont des signes ou symboles qui nous permettent d'effectuer des opérations mathématiques. Dans le langage Python, certains opérateurs sont :

Opération arithmétique	Opérateur dans Python
Addition	+
Soustraction	-
Division	/
Multiplication	*
Exposant	**
Notation scientifique ($\times 10^n$)	E ou e
Modulo (le restant d'une division)	%
Division entière (la partie entière du quotient d'une division)	//

- En programmation, les structures conditionnelles permettent d'exécuter différentes parties du code selon des conditions spécifiques.
- Les opérateurs de comparaison sont utilisés dans une condition afin de vérifier si elle est vraie ou fausse. Les 6 opérateurs de base sont : ==, !=, <, >, ≤, ≥.

Déroulement pour l'exemple 1

- Consulter, au besoin, la fiche **Les concepts de base de programmation** de la section **Concepts mathématiques** afin de revoir avec les élèves les concepts de base en programmation ainsi que la terminologie liée à ces concepts en vue de les aider à réaliser l'activité.
- 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 :
 - Quelle est l'importance de l'utilisation de la programmation pour résoudre des problèmes mathématiques?
 - Quelles sont les ressemblances et les différences entre les différents langages de programmation que tu as déjà utilisés?
 - Quels sont les différents langages de programmation?
- Présenter aux élèves l'exemple 1, soit La priorité des opérations en programmation.
- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre des notions de base en programmation avec Python.
- 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 vérifier des opérations arithmétiques simples.

Notes pédagogiques

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

EXEMPLE 1

La priorité des opérations en programmation

Vérifie les équations suivantes en utilisant un environnement de programmation de ton choix.

a) $4 \times (2 + 4) \div 2 + 2 = 14$

b) $4 \times 2 + 4 \div 2 + 2 = 12$

c) $4 \times 2 + 4 \div (2 + 2) = 9$

d) $4 \times (2 + 4) \div (2 + 2) = 6$

Corrigé

Notes pédagogiques

- L'élève sélectionne la stratégie pour résoudre le problème et utilise l'outil de programmation le plus efficace pour 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.
- Il est possible que l'élève éprouve certaines difficultés à élaborer un premier programme dans le langage Python. Afin d'aider l'élève, poser les questions suivantes :
 - De quelle fonction penses-tu avoir besoin pour afficher le résultat des calculs?
 - Quels opérateurs sont nécessaires pour effectuer des calculs?
 - De quelle façon les variables sont-elles utiles pour effectuer des calculs?

J'utilise Python pour évaluer chaque expression et vérifier les équations données.



STRATÉGIE 1

En affectant le résultat à des variables

- J'affecte les calculs aux variables a , b , c et d respectivement.
- J'affiche les résultats des calculs à l'aide de la fonction `print()`.

```

1 # Programme qui vérifie la priorité des opérations dans des égalités.
2
3 a = 4 * (2 + 4) / 2 + 2 # Affecter le premier calcul à la variable "a"
4 b = 4 * 2 + 4 / 2 + 2 # Affecter le premier calcul à la variable "b"
5 c = 4 * 2 + 4 / (2 + 2) # Affecter le premier calcul à la variable "c"
6 d = 4 * (2 + 4) / (2 + 2) # Affecter le premier calcul à la variable "d"
7
8 print("La valeur de 'a)' est ", a)
9 print("La valeur de 'b)' est ", b)
10 print("La valeur de 'c)' est ", c)
11 print("La valeur de 'd)' est ", d)

```

```

La valeur de 'a)' est 14.0
La valeur de 'b)' est 12.0
La valeur de 'c)' est 9.0
La valeur de 'd)' est 6.0

```

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

 **As-tu bien commenté ton code pour qu'il soit lisible?**



STRATÉGIE 2

En utilisant le calcul au moment d'affichage

Je fais les calculs à même la fonction `print()` sans passer par l'intermédiaire des variables.

```

1 # Programme qui vérifie la priorité des opérations dans des égalités.
2
3 # Affichage des calculs pour a), b), c) et d)
4
5 print("La valeur de 'a)' est ", 4 * (2 + 4) / 2 + 2)
6 print("La valeur de 'b)' est ", 4 * 2 + 4 / 2 + 2)
7 print("La valeur de 'c)' est ", 4 * 2 + 4 / (2 + 2))
8 print("La valeur de 'd)' est ", 4 * (2 + 4) / (2 + 2))

```

```

La valeur de 'a)' est 14.0
La valeur de 'b)' est 12.0
La valeur de 'c)' est 9.0
La valeur de 'd)' est 6.0

```

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 pourquoi elles ou ils ont choisi ou non d'utiliser des variables. 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 le code et les concepts mathématiques et que les éléments de la section **Éléments à faire ressortir** sont abordés. Si nécessaire, poser des questions telles que :
 - Quel est l'avantage d'utiliser des variables pour affecter les calculs?
 - Que remarques-tu au niveau des résultats des calculs et de la priorité des opérations?
- 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 Les opérations sur les nombres en codage.
- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre diverses stratégies pour utiliser les différents opérateurs dans le langage Python.
- 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 afin d'effectuer des opérations arithmétiques sur des nombres plus complexes et de trouver si chaque résultat est inférieur ou supérieur à 0.

Notes pédagogiques

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

EXEMPLE 2

Les opérations sur les nombres en codage

Calcule la valeur de chaque expression numérique en utilisant du code Python dans l'environnement de ton choix et indique si chaque valeur est supérieure ou inférieure à 0.

a) $4,55 \times 10^3 + 3,23 \times 10^4$

b) 5^3

c) $(-4)^{-3}$

d) $\frac{(-5)^6}{(-5)^4}$

Corrigé

Notes pédagogiques

- Il est possible que l'élève utilise une autre notation pour l'expression a). Python offre un raccourci pour la notation scientifique en utilisant l'opérateur « E », mais l'élève peut également choisir d'écrire l'expression en utilisant le « $\times 10^{**3}$ ».
 - Puisqu'il y a des comparaisons à faire pour chaque calcul, l'utilisation des variables est encouragée.
 - Faire ressortir le fait que la structure conditionnelle se répète 4 fois dans le programme et expliquer à l'élève que les langages de programmation offrent une façon de regrouper ce code dans une autre structure appelée une fonction. Ceci permettra l'élève de faire des liens lors de ces apprentissages dans une mini-leçon subséquente.
- J'affecte les calculs aux variables a, b, c et d respectivement.
 - Je fais les comparaisons avec if...else : pour chacun des calculs et j'affiche un message approprié selon le résultat avec la fonction print().

```

1 # Programme qui calcule le résultat de certaines expressions.
2
3 a = 4.55E3 + 3.23E4 # Affecter le premier calcul à la variable "a"
4 b = 5**3 # Affecter le premier calcul à la variable "b"
5 c = (-4)**-3 # Affecter le premier calcul à la variable "c"
6 d = (-5)**6/(-5)**4 # Affecter le premier calcul à la variable "d"
7
8 # Effectuer la vérification pour l'expression a)
9▼ if a<0:
10     print("La valeur de 'a)' est ",a,"et elle est inférieure à 0.")
11▼ else:
12     print("La valeur de 'a)' est ",a,"et elle est supérieure ou égale à 0.")
13
14 # Effectuer la vérification pour l'expression a)
15▼ if b<0:
16     print("La valeur de 'b)' est ",b,"et elle est inférieure à 0.")
17▼ else:
18     print("La valeur de 'b)' est ",b,"et elle est supérieure ou égale à 0.")
19
20 # Effectuer la vérification pour l'expression a)
21▼ if c<0:
22     print("La valeur de 'c)' est ",c,"et elle est inférieure à 0.")
23▼ else:
24     print("La valeur de 'c)' est ",c,"et elle est supérieure ou égale à 0.")
25
26 # Effectuer la vérification pour l'expression a)
27▼ if d<0:
28     print("La valeur de 'd)' est ",d,"et elle est inférieure à 0.")
29▼ else:
30     print("La valeur de 'd)' est ",d,"et elle est supérieure ou égale à 0.")

```

```

La valeur de 'a)' est 36850.0 et elle est supérieure ou égale à 0.
La valeur de 'b)' est 125 et elle est supérieure ou égale à 0.
La valeur de 'c)' est -0.015625 et elle est inférieure à 0.
La valeur de 'd)' est 25.0 et elle est supérieure ou égale à 0.

```

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

? Existe-t-il une façon plus efficace d'écrire ton code?

Retour sur l'exemple 2

- Demander à quelques élèves de faire part au groupe-classe de leur solution et d'expliquer les stratégies pour déterminer si chaque calcul donne une valeur supérieure ou inférieure à 0. 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 opérations sur les nombres et leur représentation avec 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 opérations sur les nombres en notation scientifique avec le codage?
- 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 Évaluation d'expressions algébriques pour une valeur de x .
- Allouer aux élèves le temps requis pour effectuer le travail. À cette étape-ci, l'élève découvre diverses stratégies pour utiliser les différents opérateurs dans le langage Python.
- 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 3

L'élève utilise un logiciel de programmation afin d'évaluer des expressions algébriques simples et de trouver la valeur maximale.

Notes pédagogiques

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

EXEMPLE 3

Évaluation d'expressions algébriques pour une valeur de x

Crée un programme qui évalue les expressions suivantes pour une valeur de x égale à 3 et détermine l'expression qui a la plus grande valeur.

- a) x
- b) $x + 4$
- c) $12x - 7$
- d) $-2x^2 + 5x^3$
- e) $x^3 \div x^3$

Corrigé

Notes pédagogiques

- L'élève doit réaliser qu'une variable qui tient compte de la valeur la plus grande est nécessaire, car elle ou il ne possède pas d'autres outils pour faire le suivi de cette valeur à ce stade de son apprentissage. Poser les questions suivantes pour l'appuyer dans sa réflexion :
 - Comment penses-tu faire le suivi de la valeur maximale des calculs des expressions?
 - De quels outils ou structures penses-tu avoir besoin pour faire la vérification de la valeur la plus grande?
- Il est possible que la vérification du résultat le plus grand puisse occasionner de la difficulté chez l'élève. Plus particulièrement avec le calcul pour a), car il n'y a pas de structure conditionnelle nécessaire pour cette valeur, comme elle est la première. Poser les questions suivantes pour appuyer l'élève :
 - Est-il nécessaire de faire une vérification pour chaque calcul pour voir si le résultat est le plus grand?
- Il est également possible d'utiliser la fonction $\max(a, b)$ dans le langage Python pour comparer deux valeurs au lieu d'une structure conditionnelle. Certains élèves peuvent poser la question et elles ou ils sont encouragés à effectuer une recherche pour trouver une façon plus efficace d'écrire son code.
- Faire ressortir le fait que la structure conditionnelle se répète 5 fois dans le programme et expliquer à l'élève que les langages de programmation offrent une façon de regrouper ce code dans une autre structure appelée une fonction. Ceci permettra l'élève de faire des liens lors de ces apprentissages dans une mini-leçon subséquente.

- J'affecte les calculs aux variables a, b, c, d et e respectivement.
- Je fais la vérification du résultat maximal en utilisant une comparaison avec une structure conditionnelle if : pour chacun des calculs.
- J'affiche le résultat de chaque calcul.
- J'affiche le résultat maximal à la fin.

```

1  # Programme qui évalue des expressions pour une valeur de x.
2
3  x = 3 # Établissement de la valeur de x.
4
5  # Calculs pour a), b), c), d), et e)
6
7  a = x
8  # Assignment de a) comme valeur maximale
9  Valeur_Max = x
10 print("La valeur de 'a)' est ",a, ".")
11
12 b = x + 4
13 # Vérifier si le résultat de b) est plus grand que a)
14 ▼ if b > a:
15     Valeur_Max = b
16     print("La valeur de 'b)' est ",b, ".")
17
18 c = 12*x - 7
19 # Vérifier si le résultat de c) est plus grand que b)
20 ▼ if c > b:
21     Valeur_Max = c
22     print("La valeur de 'c)' est ",c, ".")
23
24 d = -2*x**2 + 5*x**3
25 # Vérifier si le résultat de d) est plus grand que c)
26 ▼ if d > c:
27     Valeur_Max = d
28     print("La valeur de 'd)' est ",d, ".")
29
30 e = x**3/x**3
31 # Vérifier si le résultat de e) est plus grand que d)
32 ▼ if e > d:
33     Valeur_Max = e
34     print("La valeur de 'e)' est ",e, ".")
35
36 # Affichage du plus grand résultat
37 print("La valeur la plus grande est",Valeur_Max, ".")

```

```

La valeur de 'a)' est 3 .
La valeur de 'b)' est 7 .
La valeur de 'c)' est 29 .
La valeur de 'd)' est 117 .
La valeur de 'e)' est 1.0 .
La valeur la plus grande est 117 .

```

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

? Est-ce qu'il y a une meilleure façon de comparer deux nombres?

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 déterminer l'expression qui a la plus grande valeur lorsque $x = 3$. 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 expressions algébriques et le codage et que les éléments de la section **Éléments à faire ressortir** sont abordés. Si nécessaire, poser une question telle que : Y a-t-il des limites à l'utilisation du codage pour représenter et évaluer des expressions algébriques?
- Encourager les élèves à ajouter des éléments à leur travail ou à consigner les éléments essentiels dans leur cahier de notes.

ALLER PLUS LOIN

Crée un programme qui prend les valeurs numériques de 3 variables et qui utilise des structures conditionnelles (imbriqués) afin de les sortir en ordre décroissant.

Par exemple :

`val1 = 4`

`val2 = -10`

`val3 = 12`

Sortie : 12, 4, -10

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. Utilise le code saisi ci-dessous pour écrire un programme qui détermine si la valeur de x est paire ou impaire.

```
1 # Programme qui vérifie une valeur de "x" pour déterminer si elle est paire ou non
2
3 # Affectation de la valeur de "x" à partir du clavier.
4 x = int(input("Veuillez entrer une valeur pour 'x' svp : "))
5
6 if x % 2 != 0: # Le résultat n'est pas 0.
7     print("Le chiffre",x,"est impair.")
8 else:
9     print("Le chiffre",x,"est pair.")
```

Processus mathématique préconisé : réflexion

L'élève réfléchit à la meilleure façon de déterminer si un chiffre est pair ou impair et d'utiliser le code pour représenter cette stratégie.

J'utilise le code Python fourni qui accepte une valeur du clavier.

Je fais la vérification de pair ou impair avec l'utilisation de l'opérateur % qui donne le reste d'une division. S'il y a un un reste suite à la division par 2, le chiffre est impair, sinon il est pair.

Je fais plusieurs essais avec mon programme afin de valider son fonctionnement.

```

1 # Programme qui vérifie une valeur de "x" pour déterminer si elle est paire ou non
2
3 # Affectation de la valeur de "x" à partir du clavier.
4 x = int(input("Veuillez entrer une valeur pour 'x' svp : "))
5
6 if x % 2 != 0: # Le résultat n'est pas 0.
7     print("Le chiffre",x,"est impair.")
8 else:
9     print("Le chiffre",x,"est pair.")

```

```

Veuillez entrer une valeur pour 'x' svp : 22
Le chiffre 22 est pair.
>

```

```

Veuillez entrer une valeur pour 'x' svp : 11
Le chiffre 11 est impair.
>

```

```

Veuillez entrer une valeur pour 'x' svp : 0
Le chiffre 0 est pair.
>

```

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

- Écris un programme qui utilise les nombres 20, 10, 15 et 2 et des opérations arithmétiques avec différents regroupements pour arriver à une réponse de 95.

Processus mathématique préconisé : sélections d'outils, de technologie et de stratégies

L'élève utilise le codage pour garder des traces de ces calculs pour faire le suivi de ses combinaisons d'opérations.

Les réponses peuvent varier :

- J'assigne les valeurs de l'énoncé afin de faciliter leur utilisation.
- J'utilise Python pour faire plusieurs essais et vérifier les résultats.
- Je garde une trace de mes résultats afin de m'aider à trouver d'autres combinaisons d'opérations.

```

1 # Programme qui essaie différentes opérations arithmétiques pour
  trouver
2 # un résultat de 95 avec les valeurs de w, x, y et z.
3
4 # Affectation des valeurs de w, x, y et z.
5
6 w = 20
7 x = 10
8 y = 15
9 z = 2
10
11 # Essai 1
12 print("Essaie 1: (w+x)*y/z=", (w+x)*y/z)
13
14 # Essai 2
15 print("Essaie 2: ((w+x)*y)/z=", ((w+x)*y)/z)
16
17 # Essai 3
18 print("Essaie 3: ((w+x)*y/z)**z=", ((w+x)*y/z)**z)
19
20 # Essai 4

```

```

Essai 1: (w+x)*y/z= 225.0
Essai 2: ((w+x)*y)/z= 225.0
Essai 3: ((w+x)*y/z)**z= 50625.0
Essai 4: w+(x*y)/z= 95.0

```

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

? *Peux-tu trouver d'autres calculs qui donnent une réponse de 95?*

- Écris un programme qui utilise une valeur n (en cm) et qui calcule l'aire et le périmètre d'un carré ayant des côtés de longueur n ainsi que l'aire et la circonférence d'un cercle de rayon n .

Processus mathématique préconisé : communication

L'élève utilise les commentaires et les noms de variable pour bien expliquer son code et les calculs qu'elle ou qu'il fait.

Les réponses peuvent varier :

- J'assigne une valeur à la variable n que je peux changer n'importe quand.
- J'effectue les calculs pour l'aire, le périmètre et la circonférence et je les assigne aux variables respectives.
- Je documente mon programme et je nomme mes variables de façon claire.
- J'affiche mes résultats.

```

1  # Programme qui détermine l'aire et le périmètre d'un carré dont
   la longueur des côtés est n
2  # ainsi que l'aire et la circonférence d'un cercle de rayon n.
3
4  # Affectation de la valeur de n.
5  n = 5
6  # Affectation de la constante Pi pour faciliter les calculs.
7  Pi = 3.1416
8
9  # Calculs et affichage pour le carré
10 Perimetre_carre = 4 * n
11 Aire_carre = 4**2
12 print("Le périmètre du carré est",Perimetre_carre,"cm.")
13 # Utilisation du code \u00b pour afficher 2 comme exposant.
14 print("L'aire du carré est",Aire_carre,"cm\u00b2.")
15
16 # Calculs et affichage pour le cercle
17 Circonference_cercle = 2*Pi*n
18 Aire_cercle = Pi*n**2
19 # Utilisation de la fonction round() pour arrondir la réponse au
   centième près.
20 print("La circonférence du cercle
   est",round(Circonference_cercle,2),"cm.")
21 # Utilisation du code \u00b pour afficher 2 comme exposant.
22 print("L'aire du cercle est",round(Aire_cercle,2),"cm\u00b2.")
23

```

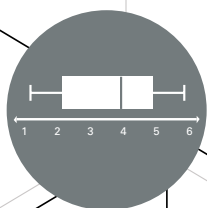
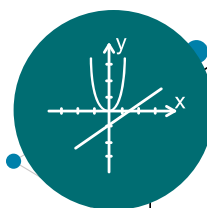
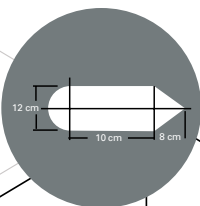
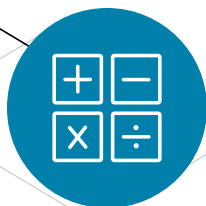
```

Le périmètre du carré est 20 cm.
L'aire du carré est 16 cm².
La circonférence du cercle est 31.42 cm.
L'aire du cercle est 78.54 cm².

```

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

Utiliser les concepts de base en programmation

Partie 1 – Exploration guidée

EXEMPLE 1

La priorité des opérations en programmation

Vérifie les équations suivantes en utilisant un environnement de programmation de ton choix.

- a) $4 \times (2 + 4) \div 2 + 2 = 14$
- b) $4 \times 2 + 4 \div 2 + 2 = 12$
- c) $4 \times 2 + 4 \div (2 + 2) = 9$
- d) $4 \times (2 + 4) \div (2 + 2) = 6$



TA STRATÉGIE

EXEMPLE 2

Les opérations sur les nombres en codage

Calcule la valeur de chaque expression numérique en utilisant du code Python dans l'environnement de ton choix et indique si chaque valeur est supérieure et inférieure à 0.

a) $4,55 \times 10^3 + 3,23 \times 10^4$

b) 5^3

c) $(-4)^{-3}$

d) $\frac{(-5)^6}{(-5)^4}$



TA STRATÉGIE

EXEMPLE 3

Évaluation d'expressions algébriques pour une valeur de x

Crée un programme qui évalue les expressions suivantes pour une valeur de x égale à 3 et détermine l'expression qui a la plus grande valeur.

- a) x
- b) $x + 4$
- c) $12x - 7$
- d) $-2x^2 + 5x^3$
- e) $x^3 \div x^3$



TA STRATÉGIE

Partie 2 – Apprentissage autonome

À ton tour!

1. Utilise le code saisi ci-dessous pour écrire un programme qui détermine si la valeur de x est paire ou impaire.

```
1 # Programme qui vérifie une valeur de "x" pour déterminer si elle est paire ou non
2
3 # Affectation de la valeur de "x" à partir du clavier.
4 x = int(input("Veuillez entrer une valeur pour 'x' svp : "))
5
6 if x % 2 != 0: # Le résultat n'est pas 0.
7     print("Le chiffre",x,"est impair.")
8 else:
9     print("Le chiffre",x,"est pair.")
```



TA STRATÉGIE

2. Écris un programme qui utilise les nombres 20, 10, 15 et 2 et des opérations arithmétiques avec différents regroupements pour arriver à une réponse de 95.



TA STRATÉGIE

3. Écris un programme qui utilise une valeur n (en cm) et qui calcule l'aire et le périmètre d'un carré ayant des côtés de longueur n ainsi que l'aire et la circonférence d'un cercle de rayon n .



TA STRATÉGIE