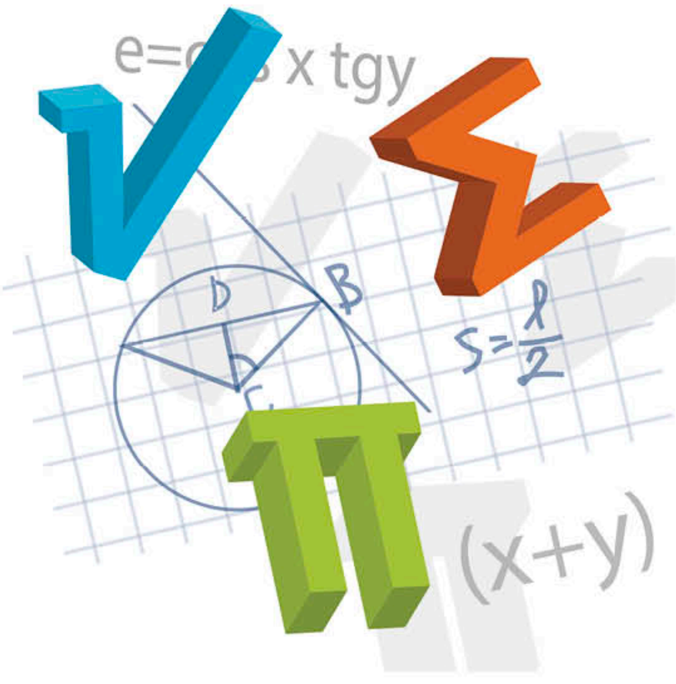
2de

Maîtriser les méthodes fondamentales de maths



ellipses

Chapitre 1

Les fondamentaux en algorithmique

I - Un peu de théorie

1) Quelques notions

a) Algorithme et notion de variable

Définition 1 : Un **algorithme** est une suite finie d'instructions à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat.

Pour stocker un résultat, on utilise une **variable**. On peut se représenter une variable comme une «boîte». Pour pouvoir accéder à son contenu, on lui donne un nom.
Elle contient une **valeur**. On utilise différents **types de valeurs** :

- **entier** (nombre entier relatif)
- **flottant** (nombre à virgule)
- **chaîne de caractères** (suite ordonnée de caractères, un caractère étant un chiffre, une lettre, un symbole,...)
- ...

b) L'affectation

Lorsque l'on donne une valeur à une variable X, on écrit l'instruction

$$X \leftarrow \text{.............}$$

La nouvelle valeur remplace la précédente.

2) Structure conditionnelle «Si»

Définition 2 : Un **booléen** est un type de variable à deux états : Vrai ou Faux.

La structure conditionnelle «Si... Alors... Sinon» permet d'écrire une instruction conditionnelle dans un algorithme.

Si {booléen B}
    Alors {instructions I_1}
    Sinon {instructions I_2}
Fin Si

On utilise ce type de structure lorsque les instructions que l'on souhaite exécuter dépendent de la valeur d'un booléen.
La valeur de B décide de l'exécution de la phase de traitement :

- si la valeur de B est Vrai, seules les instructions I_1 sont exécutées,
- si la valeur de B est Faux, seules les instructions I_2 sont exécutées.

Au lycée, le langage de programmation est Python. Voici comment s'écrit la structure conditionnelle en Python :

```
if booleen B :
  Instructions I1
else :
  Instructions I2
```

Exemple : On veut savoir si une personne est mineure ou pas, le test porte donc sur son âge.

Si $a < 18$
    Alors Afficher "La personne est mineure"
    Sinon Afficher "La personne est majeure"
Fin Si

Voici l'écriture en Python de cet algorithme :

```
if a<18 :
  print("La personne est mineure")
else :
  print("La personne est majeure")
```

Voici des exemples d'exécution de l'algorithme :

```
a=12
if a<18 :
    print("La personne est mineure")
else :
    print("La personne est majeure")
```

```
La personne est mineure
```

```
a=21
if a<18 :
    print("La personne est mineure")
else :
    print("La personne est majeure")
```

```
La personne est majeure
```

Remarque.

- Pour afficher un nombre, une chaîne de caractères,... sur Python, on utilise la fonction print.
- L'espace présent en début de ligne est appelé l'indentation. Il doit être présent sous peine d'erreur de compilation :

```
if a<18 :
print("La personne est mineure")
else :
    print("La personne est majeure")
```

```
  Cell In[1], line 2
    print("La personne est mineure")
    ^
IndentationError: expected an indented block after 'if'
statement on line 1
```

- Il ne faut pas oublier de mettre les « : » sous peine d'erreur de compilation.

```
if a<18
    print("La personne est mineure")
else :
    print("La personne est majeure")
```

```
  Cell In[5], line 1
    if a<18
           ^
SyntaxError: expected ':'
```

- Pour faire une affectation en Python, il suffit de mettre un $=$ à la place de $\leftarrow$.
- En Python, pour tester si deux variables A et B sont égales, on écrit $A == B$.
- En Python, pour tester si deux variables A et B sont différentes, on écrit $A! = B$.
- En Python, pour tester une inégalité large, on écrit $A <= B$ ou $A >= B$.
- On peut également utiliser la structure incomplète «**Si** ...**Alors** ...». Dans ce cas, on ne fait rien lorsque la condition n'est pas vérifiée.

3) Boucle bornée «Pour»

On peut répéter un bloc d'instructions un certain nombre de fois fixé au départ : on utilise alors la boucle «Pour».

Exemple : La ville de Corbeil-Essonnes compte aujourd'hui 52 000 habitants. Chaque année, cette population augmente de 300 habitants.
On souhaite élaborer un algorithme donnant le nombre d'habitants de cette ville dans N années.
Pour cela, on définit une variable P qu'on initialise à 52 000, puis on répète N fois l'opération qui consiste à ajouter 300 à P : le nombre de répétitions (ou «itérations») est connu au départ puisque c'est le nombre N d'années.

$P \leftarrow 52\,000$
Pour i variant de 1 à N
    $P \leftarrow P + 300$
Fin Pour

La boucle bornée en Python s'écrit comme suit :

```
P=52000
for i in range(N) :
  P=P+300
```

On suppose qu'on a saisi $N = 3$ en entrée.

Étapes	i	P
Avant le début de la boucle		52 000
Fin de la 1$^{\text{ère}}$ itération	0	52 300
Fin de la 2$^{\text{ème}}$ itération	1	52 600
Fin de la 3$^{\text{ème}}$ itération	2	52 900

On peut vérifier à l'aide de l'ordinateur que la valeur de P en fin d'algorithme est bien 52 900.

```
P=52000
for i in range(3) :
    P=P+300
print(P)
```

```
52900
```

Remarque. Voici les différentes utilisations de la commande range. Pour faciliter sa compréhension, on exécute les programmes suivants :

```
for i in range(4) :
    print(i)
```

```
0
1
2
3
```

```
for i in range(1,4) :
    print(i)
```

```
1
2
3
```

```
for i in range(2,5) :
    print(i)
```

```
2
3
4
```

Quand la fonction range n'a qu'un argument n, elle crée une liste contenant tous les entiers allant de 0 à $n-1$. Cette liste contient bien n éléments.
Quand la fonction range possède deux arguments n et m, alors elle crée une liste contenant tous les entiers allant de n à $m-1$.

4) Boucle non bornée «Tant que»

On peut répéter un bloc d'instructions tant qu'une condition reste vérifiée : on utilise alors la boucle «Tant que».

Exemple : Une puce est capable de réaliser un saut d'une longueur de 32 centimètres. Nous souhaitons savoir au bout de combien de sauts une puce aura parcouru une distance de 1 mètre.
On définit la variable D égale à la distance parcourue par la puce. On doit donc répéter plusieurs fois l'instruction «$D \leftarrow D+32$», sans savoir à l'avance le nombre de ces répétitions.
Pour cela, on va tester si la variable D est inférieure ou égale à 100 centimètres en début de boucle. Tant que cette condition est vérifiée, on applique les instructions de la boucle.
Pour compter le nombre de sauts, on introduit un compteur C : il est initialisé à 0. À chaque fois que la boucle est parcourue, sa valeur augmente d'une unité.

$D \leftarrow 0$
$C \leftarrow 0$
Tant que $D \leqslant 100$
    $D \leftarrow D+32$
    $C \leftarrow C+1$
Fin Tant que

La boucle non bornée s'écrit en Python comme suit :

```
D=0
C=0
while D<=100 :
  D=D+32
  C=C+1
```

C'est la boucle «Tant que» qui permet de répéter ce calcul. Cette boucle est utilisée lorsque l'on veut recommencer un même bloc d'instructions jusqu'à valider une condition de sortie donnée à l'avance.
C'est la structure itérative avec fin de boucle conditionnelle.

Étapes	D	C	Condition vérifiée
Avant le début de la boucle	0	0	Vrai
1^{er} passage dans la boucle	32	1	Vrai
$2^{\text{ème}}$ passage dans la boucle	64	2	Vrai
$3^{\text{ème}}$ passage dans la boucle	96	3	Vrai
$4^{\text{ème}}$ passage dans la boucle	128	4	Faux

On peut vérifier à l'aide de l'ordinateur que la valeur de C en fin d'algorithme est bien 4.

```
D=0
C=0
while D<=100 :
    D=D+32
    C=C+1
print(C)
```

```
4
```

C'est donc au quatrième saut que la puce dépassera pour la première fois 100 centimètres.

Remarque. Le test effectué est un booléen.

5) Les fonctions en algorithmique

- Dans un programme, il est possible d'écrire des petits programmes appelés **fonctions**.
- Une fonction est un programme qui porte un nom utilisant zéro ou une ou plusieurs variables appelées **arguments**.
- Une fonction s'écrit en Python comme ci-dessous :

```
def nom_de_la_fonction(argument1,argument2,etc) :
  instruction(s)
  return resultat
```

- L'instruction return permet de renvoyer une valeur de type entier, décimal (dit flottant), chaîne de caractères ou booléen qui peut être réutilisée dans un autre programme ou une autre fonction. Elle interrompt le programme dès qu'elle s'est exécutée.

Exemple : La fonction **somme_cubes** ci-dessous calcule et renvoie la somme $m^3 + n^3$ lorsque l'utilisateur donne les valeurs de m et de n.

```
def somme_cubes(m,n) :
  somme=m**3+n**3
  return somme
```

Remarque. En Python, pour écrire par exemple 3^3, on écrira $3 * *3$.

Si m vaut 3 et n vaut 2, l'algorithme est censé retourner $3^3 + 2^3$ soit 35. Vérifions cela à l'aide de l'ordinateur.

```
def somme_cubes(m,n) :
    somme=m**3+n**3
    return somme

print(somme_cubes(3,2))
```

```
35
```

II - Approfondissements

Exercice 1.

Choisir le nombre 9 Multiplier ce nombre par 2 Ajouter 3 au résultat

1. Quel est le résultat de cet algorithme ?
2. Écrire en Python la suite d'instructions correspondant au programme de calcul. On pourra vérifier le résultat de la question 1 à l'aide de l'ordinateur.

Exercice 2. On considère l'algorithme suivant :

$$\boxed{\begin{array}{l} A \leftarrow 7 \\ B \leftarrow A + 3 \\ A \leftarrow A \times B \end{array}}$$

Écrire les différentes valeurs que prennent A et B à chaque étape. On considère que chaque ligne du tableau correspond à une ligne de l'algorithme.

A	B

Remarque. Ce type de tableau sera appelé tableau d'exécution.

Exercice 3. On donne deux suites d'instructions S_1 et S_2, où la variable A est une chaîne de caractères.

S_1
$A \leftarrow$ "algèbre"
$A \leftarrow$ "analyse"

S_2
$A \leftarrow$ "analyse"
$A \leftarrow$ "algèbre"

1. (a) Après l'exécution de S_1, quelle est la valeur de la variable A ?
   (b) Après l'exécution de S_2, quelle est la valeur de la variable A ?
2. L'ordre dans lequel on écrit les instructions a-t-il de l'importance ? Pourquoi ?

Exercice 4.
Soient deux variables A et B.

1. Réaliser un tableau d'exécution.
2. En déduire la valeur de B en fin d'algorithme.

$A \leftarrow 5$
$B \leftarrow 2$
$A \leftarrow A \times B$
$B \leftarrow A + B.$

Exercice 5.

1. Gwendoline a écrit la variable X après l'exécution de chaque instruction ci-dessous. Deux valeurs sont fausses. Corriger son travail.

$X \leftarrow 3$
$X \leftarrow 3X$
$X \leftarrow X - 1$
$X \leftarrow X \times X$

X
3
9
7
49

2. On remplace la première instruction par $X \leftarrow a$, a étant un réel donné. Quelle est, parmi les valeurs ci-contre, celle de la variable X après l'exécution de ces instructions ?

   ☐ $3a - 1^2$ ☐ $3(a-1)^2$ ☐ $(3a-1)^2$

Exercice 6. On considère deux variables A et B qui ont respectivement pour valeurs a et b. Écrire un programme en Python permettant d'échanger A et B. On pourra vérifier si l'algorithme fonctionne à l'aide d'un tableau d'exécution.

Exercice 7. Dans un parc d'attraction, un informaticien a développé un algorithme permettant de déterminer si une personne peut monter dans un manège dont la taille minimale requise est 140cm.

Si taille> 140
    Alors afficher «La personne peut monter»
    Sinon Afficher «La personne ne peut pas monter»
Fin Si

1. Si taille = 120, quel est l'affichage ? Si taille = 145 ?
2. Réécrire l'algorithme en langage Python. On pourra le tester avec les valeurs de la question 1 à l'aide de l'ordinateur.

Exercice 8. On considère l'algorithme ci-dessous :

```
if n%3==0 :
  M=n/3
else :
  M=0
```

Quelle est la valeur de la variable M pour :

1. $n = 15$
2. $n = 4$
3. $n = -5$

On pourra vérifier les résultats à l'aide de l'ordinateur.

Remarque. En Python, $n\%p$ est le reste de la division euclidienne de n par p. On a donc, par exemple,

```
5%3
```

```
2
```

En effet, le reste de la division euclidienne de 5 par 3 est 2.

Exercice 9. Un groupe de personnes souhaite réserver une maison à la plage pendant les vacances d'été. Le prix de la location à la semaine est de 900 euros. Une école de surf propose un stage d'initiation. Le tarif du stage pour une personne est de 250 euros. Cependant, il existe un tarif de groupe : si plus de 10 personnes s'inscrivent, le tarif du stage pour une personne est de 210 euros. Le groupe loue la maison et fait le stage de surf.

1. (a) Quel est le prix payé par un groupe de 4 personnes pour la semaine ?
   (b) Quel est le prix payé par un groupe de 12 personnes pour la semaine ?

2. Compléter l'algorithme ci-dessous afin que la variable P contienne en fin d'algorithme le prix payé par le groupe pour la semaine, selon le nombre N de personnes du groupe.

```
if N>=10 :
  .....................................
else :
  .....................................
```

On pourra tester l'algorithme avec les valeurs de la question 1 à l'aide de l'ordinateur.

`Remarque.` En Python, il est impossible d'écrire $3x$. Le logiciel ne comprend pas la multiplication implicite présente entre 3 et x.

```
x=4
```

```
3x
```

```
  Cell In[5], line 1
    3x
    ^
SyntaxError: invalid decimal literal
```

```
3*x
```

```
12
```

Exercice 10. On donne l'algorithme suivant :

```
A=50
for i in range(4) :
  A=A+10
```

Compléter le tableau ci-dessous et en déduire la valeur de A à la fin de l'algorithme.

Étapes	i	A
Avant le début de la boucle		
Fin de la 1$^{\text{ère}}$ itération	0	
...	...	...

On pourra vérifier le résultat à l'aide de l'ordinateur.

Exercice 11. Mme Lassel effectue un unique virement de 10 000 euros sur un livret d'épargne. Le capital est augmenté chaque année de 200 euros par le versement d'intérêts.

1. Déterminer le capital disponible au bout d'un an sur le livret d'épargne de Mme Lassel.
2. Compléter l'algorithme ci-contre pour que la valeur de capital à la fin de l'algorithme soit le capital disponible sur le livret au bout de 7 ans.

```
capital=..........
for i in range(.....) :
  capital=................
```

3. À l'aide d'un tableau d'exécution, que vaut le capital de Mme Lassel au bout de 7 ans ? On pourra vérifier le résultat à l'aide de l'ordinateur.

Exercice 12. Le 1er janvier 2024, Théo a reçu 50 euros d'étrennes, puis chaque année, celles-ci augmentent de 5 euros. Théo décide de ne pas dépenser cet argent avant de pouvoir s'acheter la PS4 qui coûte 450 euros.

1. Déterminer le montant E des étrennes que Théo reçoit le 1er janvier 2025.
2. En déduire la somme totale T dont il dispose en 2025.
3. L'algorithme ci-contre permet de calculer l'année durant laquelle Théo pourra s'acheter la PS4.

```
E=50
T=50
A=2024
while T<450 :
  E=E+5
  T=T+E
  A=A+1
```

   (a) Construire un tableau permettant de déterminer les valeurs successives de E, de T, de A et permettant de savoir si la condition de boucle est vérifiée.

   (b) En déduire à partir de quelle année, Théo pourra dépenser son argent. On pourra vérifier le résultat à l'aide de l'ordinateur.

Exercice 13. On a écrit le script de la fonction **exercice**.

```
def exercice(a) :
  return a**2-3*a+5
```

1. Combien de paramètres cette fonction possède-t-elle ? Le(s)quel(s) ?
2. Que renvoie la fonction lorsqu'on saisit les instructions suivantes ?

   (a) **exercice(3)** (b) **exercice(**-1**)** (c) **exercice(3.2)**

Exercice 14. On propose la fonction suivante.

```
from random import *
def exercice() :
  return randint(1,6)
```

1. Quelle est l'utilité de l'instruction de la première ligne ?
2. Combien de paramètres possède cette fonction ?
3. Quelles sont les valeurs possibles de **exercice()** ? On pourra s'aider de l'ordinateur.
4. Proposer un cas concret dans lequel cette fonction pourrait avoir un intérêt.

Exercice 15. La fonction **cercle** ci-dessous calcule et renvoie l'aire d'un disque lorsqu'on entre son rayon R.

```
from math import pi
def cercle(R) :
  return pi*R**2
```

1. À quoi sert l'instruction de la première ligne ?
2. Que vaut l'aire d'un disque de rayon 2 ? *On arrondira à* 10^{-3} *près.*
3. Copier le script de la fonction **cercle** dans l'éditeur Python puis le modifier pour que le résultat renvoyé soit arrondi à 10^{-3} près. Vérifier si l'algorithme fonctionne avec la valeur de la question 2 à l'aide de l'ordinateur.

Remarque. Afin de déterminer l'arrondi d'un nombre, on peut utiliser la fonction **round(nombre à approcher,nombre de chiffres après la virgule)**.

Exercice 16.

1. Copier le script de la fonction **repetition** dans l'éditeur de Python.

```
def repetition(n) :
  chaine=""
  for i in range(n) :
    chaine=chaine+"bla"
  return chaine
```

2. Taper l'instruction **repetition(7)** dans la console Python. Quel est le résultat affiché ?
3. Que fait l'opérateur «+» quand les termes sont des chaînes de caractères ?

Exercice 17.

1. Écrire une fonction **solde** qui renvoie le prix soldé d'un article lorsque l'utilisateur donne le prix initial p (en euro) et la remise t (en pourcentage).
2. Un article de 25 euros subit une diminution de 40% par rapport au prix initial. Quel est son nouveau prix ? On pourra utiliser l'algorithme de la question précédente.

III - Correction des approfondissements

Exercice 1.

1. On prend le nombre 9 qu'on multiplie par 2. On obtient donc 18 auquel on ajoute 3. Par conséquent, le résultat de cet algorithme est 21.
2. On peut utiliser une variable A.

```
A=9
A=2*A
A=A+3
```

À l'aide de l'ordinateur, on obtient bien le même résultat qu'à la question 1.

```
A=9
A=A*2
A=A+3
print(A)
```

```
21
```

Exercice 2. On a

A	B
7	
7	10
70	10

Exercice 3.

1. (a) On peut faire un tableau d'exécution :

A
algèbre
analyse

Ainsi, A prend la valeur analyse après l'exécution de S_1.

(b) On peut faire un tableau d'exécution :

A
analyse
algèbre

Ainsi, A prend la valeur algèbre après l'exécution de S_2.

2. L'ordre dans lequel sont écrites les instructions a de l'importance. En effet, la seule différence entre les deux algorithmes est l'ordre. Cependant, le résultat est différent.

Exercice 4.

1. On a

A	B
5	
5	2
10	2
10	12

2. Par conséquent, B prend la valeur 12 en fin d'algorithme.

Exercice 5.

1. Voici la correction du travail de Gwendoline :

X
3
9
8
64

2. On peut réaliser un tableau d'exécution.

X
a
$3a$
$3a-1$
$(3a-1)^2$

La bonne réponse est $(3a-1)^2$.

Exercice 6. On peut introduire une troisième variable C. On a donc

```
A=a
B=b
C=A
A=B
B=C
```

On peut faire un tableau d'exécution pour vérifier si cela fonctionne.

A	B	C
a		
a	b	
a	b	a
b	b	a
b	a	a

Les valeurs des variables A et B sont bien échangées.

Exercice 7.

1. On sait que $120 \leqslant 140$. L'affichage sera donc «La personne ne peut pas monter». On sait que $145 > 140$. L'affichage sera donc «La personne peut monter».
2. On a

```
if taille>140 :
  print("La personne peut monter")
else :
  print("La personne ne peut pas monter")
```

Ainsi, en utilisant l'ordinateur, on a :

```
taille=120
if taille>140 :
    print("La personne peut monter")
else :
    print("La personne ne peut pas monter")
```

```
La personne ne peut pas monter
```

```
taille=145
if taille>140 :
    print("La personne peut monter")
else :
    print("La personne ne peut pas monter")
```

```
La personne peut monter
```

Exercice 8.

1. 15 est divisible par 3 donc $M = 5$.
2. 4 n'est pas divisible par 3 donc $M = 0$.
3. -5 n'est pas divisible par 3 donc $M = 0$.

En utilisant l'ordinateur, on a :

```
n=15
if n%3==0 :
    M=n/3
else :
    M=0
print(M)
```

```
5.0
```

```
n=4
if n%3==0 :
    M=n/3
else :
    M=0
print(M)
```

```
0
```

```
n=-5
if n%3==0 :
    M=n/3
else :
    M=0
print(M)
```

```
0
```

Exercice 9.

1. (a) Comme le groupe n'est que de 4 personnes, il ne bénéficie pas du tarif de groupe. Le stage de surf sera donc de 250 euros par personne soit 1 000 euros. À cela, on ajoute le prix de la location qui est de 900 euros. Par conséquent, le prix payé par un groupe de 4 personnes est 1 900 euros.

   (b) Comme le groupe est composé de 12 personnes, il bénéficie du tarif de groupe. Le stage de surf sera donc de 210 euros par personne soit 2 520 euros. On ajoute ensuite le prix de la location. Par conséquent, le prix payé par un groupe de 12 personnes est 3 420 euros.

2. Voici l'algorithme complété :

```
if N>=10 :
  P=900+N*210
else :
  P=900+N*250
```

En utilisant l'ordinateur, on a

```
N=4
if N>=10 :
    P=900+N*210
else :
    P=900+N*250
print(P)
```

```
1900
```

```
N=12
if N>=10 :
    P=900+N*210
else :
    P=900+N*250
print(P)
```

```
3420
```

Exercice 10. On a

	i	A
Avant le début de la boucle		50
Fin de la 1ère itération	0	60
Fin de la 2ème itération	1	70
Fin de la 3ème itération	2	80
Fin de la 4ème itération	3	90

Ainsi, la valeur de A en fin d'algorithme sera 90. À l'aide de l'ordinateur, on obtient bien le même résultat.

```
A=50
for i in range(4) :
    A=A+10
print(A)
```

```
90
```

Exercice 11.

1. Un an plus tard, le capital de Mme Lassel aura été augmenté de 200 euros. Elle aura donc sur son compte en banque 10 200 euros.
2. On a

```
capital=10000
for i in range(7) :
  capital=capital+200
```

3. On a

Étapes	i	capital
Avant le début de la boucle		10 000
1er passage dans la boucle	0	10 200
2ème passage dans la boucle	1	10 400
3ème passage dans la boucle	2	10 600
4ème passage dans la boucle	3	10 800
5ème passage dans la boucle	4	11 000
4ème passage dans la boucle	5	11 200
5ème passage dans la boucle	6	11 400

À l'aide de l'ordinateur, on obtient bien le même résultat.

```
capital=10000
for i in range(7) :
    capital=capital+200
print(capital)
```

```
11400
```

Exercice 12.

1. Le montant des étrennes de Théo le 1er janvier 2025 est 55 euros.
2. En 2025, Théo disposera alors de 105 euros.
3. (a) On a

Étapes	E	T	A	Condition vérifiée
Avant le début de la boucle	50	50	2024	Vrai
1er passage dans la boucle	55	105	2025	Vrai
2ème passage dans la boucle	60	165	2026	Vrai
3ème passage dans la boucle	65	230	2027	Vrai
4ème passage dans la boucle	70	300	2028	Vrai
5ème passage dans la boucle	75	375	2029	Vrai
6ème passage dans la boucle	80	455	2030	Faux

   (b) Théo pourra dépenser son argent en 2030. À l'aide de l'ordinateur, on obtient bien le même résultat.

```
E=50
T=50
A=2024
while T<450 :
    E=E+5
    T=T+E
    A=A+1
print(A)
```

```
2030
```

Exercice 13.

1. La fonction possède un paramètre : a.
2. (a) **exercice(3)** renvoie 5.
   (b) **exercice(−1)** renvoie 9.
   (c) **exercice(3.2)** renvoie 5,64.

   En utilisant l'ordinateur, on a :

```
def exercice(a) :
    return a**2-3*a+5

print(exercice(3))
print(exercice(-1))
print(exercice(3.2))
```

```
5
9
5.640000000000001
```

Exercice 14.

1. La première ligne sert à importer toutes les commandes figurant dans la bibliothèque random.
2. Cette fonction ne possède aucun paramètre.
3. Les valeurs prises sont 1, 2, 3, 4, 5 et 6.
4. Cette fonction peut avoir un intérêt pour simuler un lancer de dé.

Exercice 15.

1. La première ligne sert à importer la variable pi de la bibliothèque math.
2. L'aire d'un disque de rayon 2 est de 4π soit $12,566$ à 10^{-3} près.
3. On a

```
from math import pi

def cercle(R) :
  return round(pi*R**2,3)
```

   On obtient bien le résultat de la question 2 en utilisant l'ordinateur.

```
from math import pi

def cercle(R) :
    return round(pi*R**2,3)

print(cercle(2))
```

```
12.566
```

Exercice 16.

1. et 2.

```
def repetition(n) :
    chaine=""
    for i in range(n) :
        chaine=chaine+"bla"
    return chaine
```

```
repetition(7)
```

```
'blablablablablablabla'
```

3. L'opérateur «+» permet de créer une nouvelle chaîne de caractères dans laquelle on met bout à bout les deux chaînes. Cela s'appelle la concaténation.

Exercice 17.

1. On a

```
def solde(p,t) :
  return p-p*t/100
```

2. On utilise l'ordinateur :

```
def solde(p,t) :
    return p-p*t/100

print(solde(25,40))
```

```
15.0
```

Ainsi, le nouveau prix de l'article à 25 euros après une remise de 40% est de 15 euros.

Chapitre 2

Ensemble de nombres

I - Un peu de théorie

1) Nombres entiers

Définition 1 :

- L'ensemble des **entiers naturels** est

$$\mathbb{N} = \{0; 1; 2; 3; ...\}.$$

- L'ensemble des **entiers relatifs** est

$$\mathbb{Z} = \{..., -2; -1; 0; 1; 2; ...\}.$$

Définition 2 : Soient A et B deux ensembles. On dit que A **est inclus dans** B si tout élément de A est un élément de B. On le note $A \subset B$.

2) L'ensemble des nombres décimaux : $\mathbb{D}$

Définition 3 : Un **nombre décimal** est un nombre qui peut s'écrire sous la forme $\frac{a}{10^p}$, avec a un entier relatif et p un entier naturel.

`Remarque.`

- Un nombre décimal peut s'écrire avec un nombre fini de chiffres après la virgule.
- Pour dire qu'un nombre appartient ou non à un ensemble, on utilise les symboles suivants : $\in$ ("appartient") et $\notin$ ("n'appartient pas").

Exemple : $\frac{1}{5}$, $0,3$ et 1 sont des nombres décimaux.

3) L'ensemble des nombres rationnels : $\mathbb{Q}$

Définition 4 : Un **nombre rationnel** est un nombre qui peut s'écrire sous la forme d'un quotient $\dfrac{a}{b}$ avec a un entier relatif et b un entier relatif non nul.
Exemple : $\dfrac{1}{3}$, $2 = \dfrac{2}{1}$ et $0,5 = \dfrac{1}{2}$ sont des nombres rationnels.

4) L'ensemble des nombres réels : $\mathbb{R}$

Définition 5 : Un nombre est **réel** s'il peut être l'abscisse d'un point d'une droite graduée appelée **droite numérique**.

Remarque. $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{D} \subset \mathbb{Q} \subset \mathbb{R}$.

Définition 6 : Un nombre réel qui n'est pas rationnel est dit **irrationnel**.
Exemple : $\sqrt{2}$ et π sont des nombres irrationnels.

5) Intervalles de $\mathbb{R}$

a) Notations

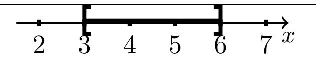
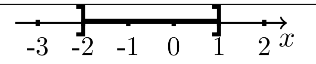
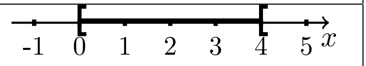
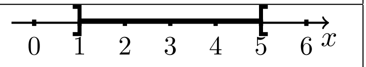
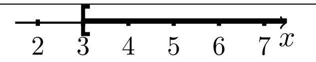
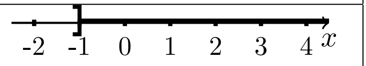
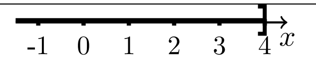
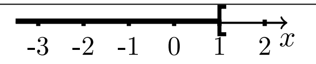
Définition 7 : **Résoudre une inéquation** c'est trouver les valeurs possibles du nombre inconnu telles que l'inégalité soit vraie.
Définition 8 : Soient a et b deux nombres réels tels que $a < b$.

- On appelle **intervalle fermé** $[a; b]$ l'ensemble des nombres réels x tels que $a \leqslant x \leqslant b$.
- On appelle **intervalle ouvert** $]a; b[$ l'ensemble des nombres réels x tels que $a < x < b$.
- On définit de même les intervalles $[a; b[$ et $]a; b]$.
- On note $[a; +\infty[$ l'ensemble des nombres réels x tels que $x \geqslant a$.
- On note $]a; +\infty[$ l'ensemble des nombres réels x tels que $x > a$.
- On définit de même $]-\infty; a]$ et $]-\infty; a[$.

Remarque.

- Le signe ∞ se lit «infini» et n'est pas un nombre.
- L'ensemble des nombres réels $\mathbb{R}$ est un intervalle qui peut se noter $]-\infty; +\infty[$.
- On note $\mathbb{R}_+ = [0; +\infty[$, $\mathbb{R}_- =]-\infty; 0]$ et $\mathbb{R}_+^* =]0; +\infty[$.

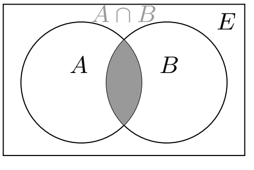
Exemple (Les différentes écritures et représentations d'un intervalle) **:**

Intervalle écrit sous forme d'inégalités	Intervalle écrit avec des crochets	Intervalle représenté sur la droite des réels
$3 \leqslant x \leqslant 6$	$[3;6]$	
$-2 < x \leqslant 1$	$]-2;1]$	
$0 \leqslant x < 4$	$[0;4[$	
$1 < x < 5$	]1 ;5[	
$x \geqslant 3$	$[3;+\infty[$	
$x > -1$	$]-1;+\infty[$	
$x \leqslant 4$	$]-\infty;4]$	
$x < 1$	$]-\infty;1[$	

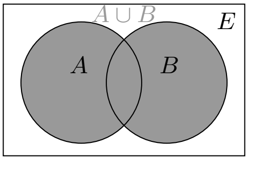
b) Intersection et réunion d'intervalles

Définition 9 : Soient A et B deux sous-ensembles de E.

• L'**intersection** de deux ensembles A et B est l'ensemble des éléments qui appartiennent à A **et** à B et se note $A \cap B$.



• La **réunion** de deux ensembles A et B est l'ensemble des éléments qui appartiennent à A **ou** à B et se note $A \cup B$.



6) Valeur absolue

a) Définition

Définition 10 : La **valeur absolue** d'un nombre x est égale au nombre x si x est positif, et au nombre $-x$ si x est négatif.
La valeur absolue de x se note $|x|$.

`Remarque.`

- La valeur absolue d'un nombre est toujours positive.
- La valeur absolue de x est la distance entre le point A d'abscisse x et l'origine de la droite numérique.

Exemple :

⋆ La valeur absolue de 8 est 8.

⋆ Pour déterminer la valeur absolue de -5, on commence par remarquer que $-5 < 0$. On a donc $|-5| = -(-5) = 5$.

b) Distance et valeur absolue

Propriété 1 : Soient a et b deux nombres réels. Sur une droite graduée, la distance entre les points A et B d'abscisse respective a et b est le nombre $|a - b|$.

Propriété 2 : Soient a un réel et r un réel strictement positif. Dire que x est tel que $|x - a| \leqslant r$ signifie que x appartient à l'intervalle $[a - r; a + r]$.

Propriété 3 : Soient a un réel et r un réel strictement positif. Dire que x est tel que $|x - a| < r$ signifie que x appartient à l'intervalle $]a - r; a + r[$.

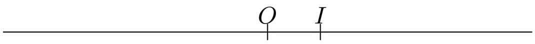
II - Un peu de pratique

1) Placer un point sur la droite des réels

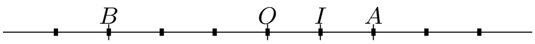
Placer sur la droite des réels les points $A(2)$ et $B(-3)$.

1. On trace une droite.

2. On place sur cette droite l'origine O et le point I qui donnera l'unité ($OI = 1$).



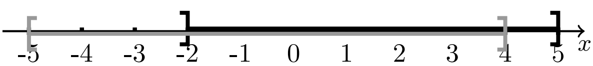
3. Si l'abscisse du point $A(a)$ est positive, on place le point A à droite de O de sorte que $OA = a$.
Si l'abscisse du point $B(b)$ est négative, on place le point B à gauche de O de sorte que $OB = -b$.



2) Déterminer la réunion et l'intersection de deux intervalles

- **Soient $I = [-5; 4]$ et $J =]-2; 5]$ deux intervalles de $\mathbb{R}$. Déterminer $I \cap J$ et $I \cup J$.**

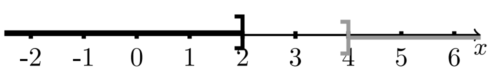
  1. On représente sur une même droite graduée les intervalles I (en gris) et J (en noir) de deux couleurs différentes.

  2. L'intersection est l'ensemble des nombres appartenant aux deux intervalles. Graphiquement, cela correspond à l'ensemble des points de la droite où les deux couleurs se superposent. Par conséquent, on a $I \cap J =]-2; 4]$.
  3. La réunion est l'ensemble des nombres appartenant à I ou à J (éventuellement les deux). Graphiquement, cela correspond à l'ensemble des points de la droite où au moins l'une des deux couleurs est présente. Par conséquent, on a $I \cup J = [-5; 5]$.

- **Soient $I =]-\infty; 2]$ et $J =]4; +\infty[$ deux intervalles de $\mathbb{R}$. Déterminer $I \cap J$ et $I \cup J$.**

  1. On représente sur une même droite graduée les intervalles I (en noir) et J (en gris) de deux couleurs différentes.

  2. Ici, aucun point ne satisfait la condition de l'intersection. On a $I \cap J = \varnothing$ et $I \cup J =]-\infty; 2] \cup]4; +\infty[$.
  Le symbole $\varnothing$ correspond à l'ensemble vide.

- **Soient $I = [-2;3]$ et $J = [3;8[$ deux intervalles de $\mathbb{R}$. Déterminer $I \cap J$ et $I \cup J$.**
    1. On représente sur une même droite graduée les intervalles I (en noir) et J (en gris) de deux couleurs différentes.

    2. Ici, seul le point d'abscisse 3 satisfait la condition de l'intersection. On note $I \cap J = \{3\}$ et $I \cup J = [-2;8[$.

3) Résoudre une inéquation du premier degré à une inconnue

- **Résoudre sur $\mathbb{R}$ l'inéquation $3x < 6$.**
    1. Le but est d'isoler x.

$$\begin{aligned} 3x < 6 &\iff \frac{1}{3} \times 3x < \frac{1}{3} \times 6 \quad (*) \\ &\iff x < 2 \end{aligned}$$

    $(*)$: l'inégalité ne change pas de sens car $\dfrac{1}{3} > 0$.
    Le symbole $\Leftrightarrow$ signifie «est équivalent à».

    2. On écrit maintenant l'ensemble solution : $\mathcal{S} =]-\infty;2[$.
- **Résoudre sur $\mathbb{R}$ l'inéquation $-4x + 1 \geqslant -11$.**
    1. Le but est d'isoler x.

$$\begin{aligned} -4x + 1 \geqslant -11 &\iff -4x + 1 - 1 \geqslant -11 - 1 \quad (*) \\ &\iff -4x \geqslant -12 \\ &\iff -\frac{1}{4} \times (-4x) \leqslant -\frac{1}{4} \times (-12) \quad (**) \\ &\iff x \leqslant 3 \end{aligned}$$

    $(*)$: l'inégalité ne change pas de sens quand on soustrait un nombre.
    $(**)$: l'inégalité change de sens car $-\dfrac{1}{4} < 0$.

    2. On écrit l'ensemble solution : $\mathcal{S} =]-\infty;-3]$.

4) Encadrer une expression donnée à partir d'un encadrement de x

- **Si $x \in [-3; 4]$, déterminer un encadrement de $2x + 7$.**
  1. On traduit l'intervalle écrit avec des crochets sous forme d'inégalités : $x \in [-3; 4] \Leftrightarrow -3 \leqslant x \leqslant 4$.
  2. On analyse l'enchaînement de l'expression. Dans notre exemple, x est d'abord multiplié par 2 puis on ajoute 7.
  3. On applique les opérations dans l'ordre trouvé précédemment. On multiplie par 2.

     $$\begin{aligned} -3 \leqslant x \leqslant 4 &\iff -3 \times 2 \leqslant x \times 2 \leqslant 4 \times 2 \ (*) \\ &\iff -6 \leqslant 2x \leqslant 8 \end{aligned}$$

     $(*)$: les inégalités ne changent pas de sens car $2 > 0$.
     On ajoute ensuite 7.

     $$\begin{aligned} -6 \leqslant 2x \leqslant 8 &\iff -6 + 7 \leqslant 2x + 7 \leqslant 8 + 7 \ (**) \\ &\iff 1 \leqslant 2x + 7 \leqslant 15 \end{aligned}$$

     $(**)$: les inégalités ne changent pas de sens quand on ajoute un nombre.
  4. On réécrit cet encadrement sous forme d'intervalle c'est-à-dire $(2x + 7) \in [1; 15]$.
- **Si $x \in [-2; 3[$, déterminer un encadrement de $-3x - 1$.**
  1. On traduit l'intervalle écrit avec des crochets sous forme d'inégalités : $x \in [-2; 3[\iff -2 \leqslant x < 3$.
  2. On analyse l'enchaînement de l'expression. Dans notre exemple, x est d'abord multiplié par -3 puis on retire 1.
  3. On applique les opérations dans l'ordre trouvé précédemment. On multiplie par -3.

     $$\begin{aligned} -2 \leqslant x < 3 &\iff -2 \times (-3) \geqslant x \times (-3) > 3 \times (-3) \ (*) \\ &\iff 6 \geqslant -3x > -9 \end{aligned}$$

     $(*)$: les inégalités changent de sens car $-3 < 0$.
     On retire ensuite 1.

     $$\begin{aligned} 6 \geqslant -3x > -9 &\iff 6 - 1 \geqslant -3x - 1 > -9 - 1 \ (**) \\ &\iff 5 \geqslant -3x - 1 > -10 \\ &\iff -10 < -3x - 1 \leqslant 5 \end{aligned}$$

     $(**)$: les inégalités ne changent pas de sens quand on soustrait un nombre.

4. On réécrit cet encadrement sous forme d'intervalle c'est-à-dire $(-3x+1) \in]-10;5]$.

5) Déterminer la distance entre deux points d'abscisse donnée

Soient A et B deux points d'abscisse respective -3 et 4. Déterminer la distance entre les points A et B.

1. On identifie a et b. Ici, on a $a=-3$ et $b=4$.
2. On applique la propriété 1. Ici on a

$$|a-b| = |(-3)-4| = |-7| = 7.$$

3. On conclut. La distance entre A et B est 7.

6) Résoudre une équation contenant une valeur absolue

- **Résoudre sur $\mathbb{R}$ l'équation $|x+4| = -4$.**
  Une valeur absolue ne peut pas être négative. On a donc

$$\mathcal{S} = \varnothing.$$

- **Résoudre sur $\mathbb{R}$ l'équation $|3x+4| = 0$.**
  Le seul nombre réel dont la valeur absolue est nulle est 0. On a donc

$$\begin{aligned} 3x+4=0 &\iff 3x=-4 \\ &\iff x=-\frac{4}{3} \end{aligned}$$

  Par conséquent, on a

$$\mathcal{S} = \left\{-\frac{4}{3}\right\}.$$

- **Résoudre sur $\mathbb{R}$ l'équation $|3x+4| = 3$.**
  On rappelle que $|A| = A$ si A est positif et $|A| = -A$ si A est négatif. Ainsi, on a

$$\begin{aligned} |3x+4|=3 &\iff 3x+4=-3 \text{ ou } 3x+4=3 \\ &\iff 3x=-7 \text{ ou } 3x=-1 \\ &\iff x=-\frac{7}{3} \text{ ou } x=-\frac{1}{3} \end{aligned}$$

  Par conséquent, on a

$$\mathcal{S} = \left\{-\frac{7}{3};-\frac{1}{3}\right\}.$$

7) Trouver l'ensemble des nombres x vérifiant $|x-a| \leqslant r$ ou $|x-a| < r$

- **Résoudre sur $\mathbb{R}$ l'inéquation $|x-1| \leqslant 2$.**
  1. On commence par identifier a et r. Dans cet exemple, on a $a = 1$ et $r = 2$.
  2. On applique la propriété 2. On a donc

$$\mathcal{S} = [1-2; 1+2] = [-1; 3]$$

- **Résoudre sur $\mathbb{R}$ l'inéquation $|x+4| < 3$.**
  1. On identifie a et r. Dans cet exemple, on a $a = -4$ et $r = 3$.
  2. On applique la propriété 3. On a donc

$$\mathcal{S} =]-4-3; -4+3[=]-7; -1[.$$

8) Trouver l'ensemble des nombres vérifiant $|x-a| > r$

- **Résoudre sur $\mathbb{R}$ l'inéquation**

$$|x-1| > 3 \qquad (*)$$

  1. Pour résoudre cette inéquation dans $\mathbb{R}$, il suffit de résoudre

$$|x-1| \leqslant 3 \qquad (**).$$

  Or $|x-1| \leqslant 3$ équivaut à $x \in [1-3; 1+3]$ c'est-à-dire $x \in [-2; 4]$.
  2. Les solutions de l'inéquation $(*)$ sont les nombres qui ne sont pas solutions de $(**)$. Ainsi,

$$\mathcal{S} =]-\infty; -2[\cup]4; +\infty[.$$

- **Résoudre sur $\mathbb{R}$ l'inéquation**

$$|x+5| \geqslant 2 \qquad (*).$$

  1. Cette inéquation est équivalente à $|x-(-5)| \geqslant 2$.
  2. Pour résoudre l'inéquation $(*)$, il suffit de résoudre

$$|x-(-5)| < 2 \qquad (**).$$

  Or $|x-(-5)| < 2$ équivaut à $x \in]-5-2; -5+2[$ c'est-à-dire $x \in]-7; -3[$.
  3. Les solutions de $(*)$ sont tous les nombres réels qui ne sont pas solutions de $(**)$. On a donc

$$\mathcal{S} =]-\infty; -7] \cup [-3; +\infty[.$$