CHAPITRE 1

Initiation à l'algorithmique

Plan du chapitre

Introduction

Dans ce chapitre, nous allons découvrir l'algorithmique et le langage Python. Nous apprendrons à :

- Écrire des instructions simples que l'ordinateur peut exécuter ;
- Manipuler des variables et différents types de données ;
- Réaliser nos premiers programmes.

L'objectif sera de définir et de comprendre comment traduire une idée en programme, étape par étape.

Partie I Algorithmes en pseudo-code

A - Exemples

Exemple :

Pour son anniversaire, Camille a reçu une boîte de Lego.
Dans cette boîte, il y a la notice de construction d'un camion de pompier.
Camille suit le texte et les dessins de la notice dans l'ordre indiqué, et construit ainsi le camion.
Camille suit un algorithme que l'on peut appeler : **Construction d'un camion de pompier**.

Exemple :

1. Ouvrir le flacon en exerçant une forme pression sur le bouchon ;
2. Retirer le film protecteur ;
3. Remplir le gobelet-doseur avec de l'eau jusqu'au trait ;
4. Verser la totalité du contenu du gobelet-doseur dans le flacon ;
5. Agiter vigoureusement.

Ici nous avons l'algorithme : **Préparation d'un antibiotique**

Il existe bon nombre d'exemple d'algorithme dans la vie de tous les jours (recettes de cuisine, choisir un film à regarder...).

B - Définition

Définition 1 : Algorithme

Un **algorithme** est une suite ..... d'instructions à appliquer dans un 40 pour arriver, en un nombre fini d'étapes, à un certain résultat.

À retenir : **Écrire un algorithme** consiste à donner une méthode détaillée décrivant toutes les étapes d'une tâche à accomplir.

★★½☆☆ Exercice 1 (Calculs)

On donne le programme suivant :

1. Choisir un nombre entier ;
2. Déterminer les deux nombres entiers qui le suivent ;
3. Faire la somme de ces trois nombres ;
4. Diviser le résultat obtenu par 3

1. Quel résultat obtient-on en faisant fonctionner ce programme de calcul avec les choix initiaux suivants :

   (a) Le nombre 5 ? (b) Le nombre 12 ? (c) Le nombre 243 ?

   dotlines5
2. Que remarque-t-on ? Le démontrer.
   dotlines5

C - Variable

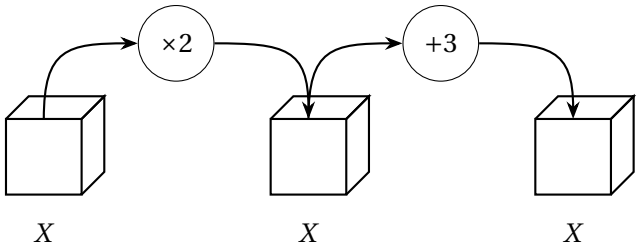
En informatique pour stocker un résultat, on utilise une variable. On peut se représenter une variable comme une « boîte » sur laquelle on colle une étiquette pour pouvoir accéder, reconnaître, son contenu.
Une variable est une zone de la mémoire de l'ordinateur dans laquelle une valeur est stockée.
Aux yeux du programmeur (nous), cette variable est définie par un nom, alors que pour l'ordinateur il s'agit en fait d'une adresse, un lieu, une zone particulière de la mémoire.

Exemple :

Considérons le programme suivant :

1. Choisir le nombre 7 ;
2. Multiplier ce nombre par 2 ;
3. Ajouter 3 au résultat obtenu.

On peut ici utiliser une variable, que l'on note X, qui contient les résultats obtenus à la fin de chaque étape du programme de calcul.



Définition 2 : Variable

Une **variable** est désignée par ....., elle contient .....
Les langages de programmation distinguent différents types de variables, une variable peut être du type :

- ..... (nombre entier relatif) ;
- ..... (nombre décimal qui représente les nombres réels) ;
- ..... (suite ordonnée de caractères, un caractère étant un chiffre, une lettre, un symbole,...) ;
- ..... (deux valeurs possibles `True` / `False`) ;
- ..... (une liste ordonnée d'objets).

Exemple :

- « Année » est une chaîne de cinq caractères. On dit que c'est une chaîne de longueur 5.
  Le premier caractère de cette chaîne est **A**, le second est **n**,...
- $L = [1,3,5,7,9]$ et $M = \left["a","e","i","o","u","y"\right]$ sont deux listes.

D - Affectation

Définition 3 : Affectation

Lorsque l'on donne une valeur à une variable X, on écrit l'instruction .....
On lit « X reçoit ... » ou « X prend la valeur... » La nouvelle valeur remplace (écrase) la valeur précédente.

★★☆☆☆ Exercice 2 (Algorithme de calcul)

Compléter le pseudo-code suivant pour traduire l'algorithme suivant :

1. Choisir le nombre 7 ;
2. Multiplier ce nombre par 2 ;
3. Ajouter 3 au résultat obtenu.

$X \leftarrow$ .....
$X \leftarrow$ .....
$X \leftarrow$ .....

★★☆☆☆ Exercice 3 (Étapes de calculs)

On considère l'algorithme suivant :

$A \leftarrow 3$
$B \leftarrow A+1$
$X \leftarrow A+B$

Donner la valeur de chacune des deux variables à chaque étape de l'algorithme :

	A	B
E1	...	
E2	...	...
E3	...	...

Pourquoi barre-t-on la casse donnant la valeur de la variable B à l'étape 1 ?

.....

★★☆☆☆ Exercice 4 (Ordre)

On se donne deux algorithmes S_1 et S_2, où la variable A est une chaîne de caractères.

S_1

$A \leftarrow$ "bonjour"
$B \leftarrow$ "bonsoir"

S_2

$A \leftarrow$ "bonsoir"
$B \leftarrow$ "bonjour"

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

   .....

   .....

★★☆☆☆ Exercice 5 (Calculs)

1. Soient A et B deux variables, considérons l'algorithme suivant :

   $A \leftarrow 1$
   $B \leftarrow 3$
   $A \leftarrow A + B$
   $B \leftarrow A \times B$

   Donner la valeur de A et B à chaque étape de l'algorithme suivant : dotlines5
2. La variable N désigne un nombre et on cherche la valeur que prend A à la fin de l'algorithme suivant :

   $N \leftarrow ...$
   $A \leftarrow N + 4$
   $A \leftarrow A \times N$
   $A \leftarrow A + 3$

   Écrire les valeurs de A et N à chaque étape dans les cas suivants :
   (a) $N \leftarrow 2$
   (b) $N \leftarrow 7$
   (c) $N \leftarrow x$

   dotlines5

★★☆☆☆ Exercice 6 (Erreurs ?)

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

Algorithme	**Juliette**	**Correction**
$X \leftarrow 2X$	4	4
$X \leftarrow X + 3$	8	8
$X \leftarrow X^2$	7 ⟵ FAUX	...
	16 ⟵ FAUX	...

   (b) On remplace la première instruction de l'algorithme par $X \leftarrow a$, a étant un réel donné.
   Entourer, parmi les valeurs ci-dessous, celle de la variable X après l'exécution de ces instructions.

   - $2a + 3^2$
   - $(2a + 3)^2$
   - $2(a + 3)^2$

2. (a) Écrire la valeur de la variable X après l'exécution de chaque instruction ci-dessous.

Algorithme
$X \leftarrow 4$
$X \leftarrow X+3$
$X \leftarrow X^2$
$X \leftarrow 2X$

Valeurs de X
...
...
...
...

(b) On remplace la première instruction par $X \leftarrow a$, a étant un réel donné.
Donner la valeur de X après l'exécution de ces instructions.

.....

.....

.....

★★½☆☆ Exercice 7 (Types)

On donne ci-dessous un algorithme :

$A \leftarrow 4$
$B \leftarrow 15$
$P \leftarrow A+B$
$P \leftarrow 2P$

1. (a) Quelles sont les variables utilisées ? .....
   (b) Quel est le type de chacune des variables ? .....
2. (a) Déterminer la valeur de P après l'exécution de la dernière instruction ?

   .....

   .....

   (b) On considère un rectangle de largeur 4cm et de longueur 15cm.
   Dans ce cas, à quoi correspond la valeur de P ? .....
3. Modifier la dernière instruction de l'algorithme afin que la valeur de P, après l'exécution de cette instruction, soit égale à la moyenne des nombres 4 et 15.

   .....

★★☆☆☆ Exercice 8 (Choix)

On se donne trois algorithmes :

A_1
$A \leftarrow Z-1$
$B \leftarrow A^2$
$C \leftarrow Z+1$
$D \leftarrow C^2$
$Z \leftarrow B+D$

A_2
$Z \leftarrow Z-1$
$A \leftarrow Z^2$
$Z \leftarrow Z+1$
$B \leftarrow Z^2$
$Z \leftarrow B+A$

A_3
$Z \leftarrow Z-1$
$B \leftarrow Z^2$
$A \leftarrow Z+2$
$C \leftarrow A^2$
$Z \leftarrow C-B$

1. Dans chaque cas, déterminer ce que contient la variable Z à la fin de l'exécution de l'algorithme lorsqu'on initialise la variable Z à la valeur 3.

   .....

   .....

.....

2. Lorsque la variable Z est initialisée à la valeur x, x étant un nombre réel, pour quel algorithme elle contient la valeur $(x-1)^2+(x+1)^2$ à la fin de l'exécution ?

.....

.....

.....

.....

E - Structure d'un algorithme

Définition 4 : Structure

Un algorithme comprend trois phases :

- Une phase ..... qui permet de donner une valeur initiale aux variables ;
- Une phase ..... ;
- Une phase .....

Une instruction de sortie d'un algorithme peut être une instruction d'affichage.

Définition 5 : Affichage

Pour **afficher** le contenu, la valeur, d'une variable X on écrit : .....

Exemple :

L'algorithme suivant :

```
X ← 4
X ← 6X
Afficher(X)
```

nous affichera : .....

★★★★★ **Exercice 9 (Échange)**

Un professeur a inversé les notes de Léa et Léo sur le bulletin de notes.
Écrire un programme qui remet les notes dans le bon ordre.
Pour cela on pourra introduire deux variables pour les notes de Léa puis de Léo, et peut-être même un troisième...
On veut que le programme affiche les résultats de la façon suivante :
Léa : *la véritable note de Léa*
Léo : *la véritable note de Léo*

.....

.....

.....

.....

.....

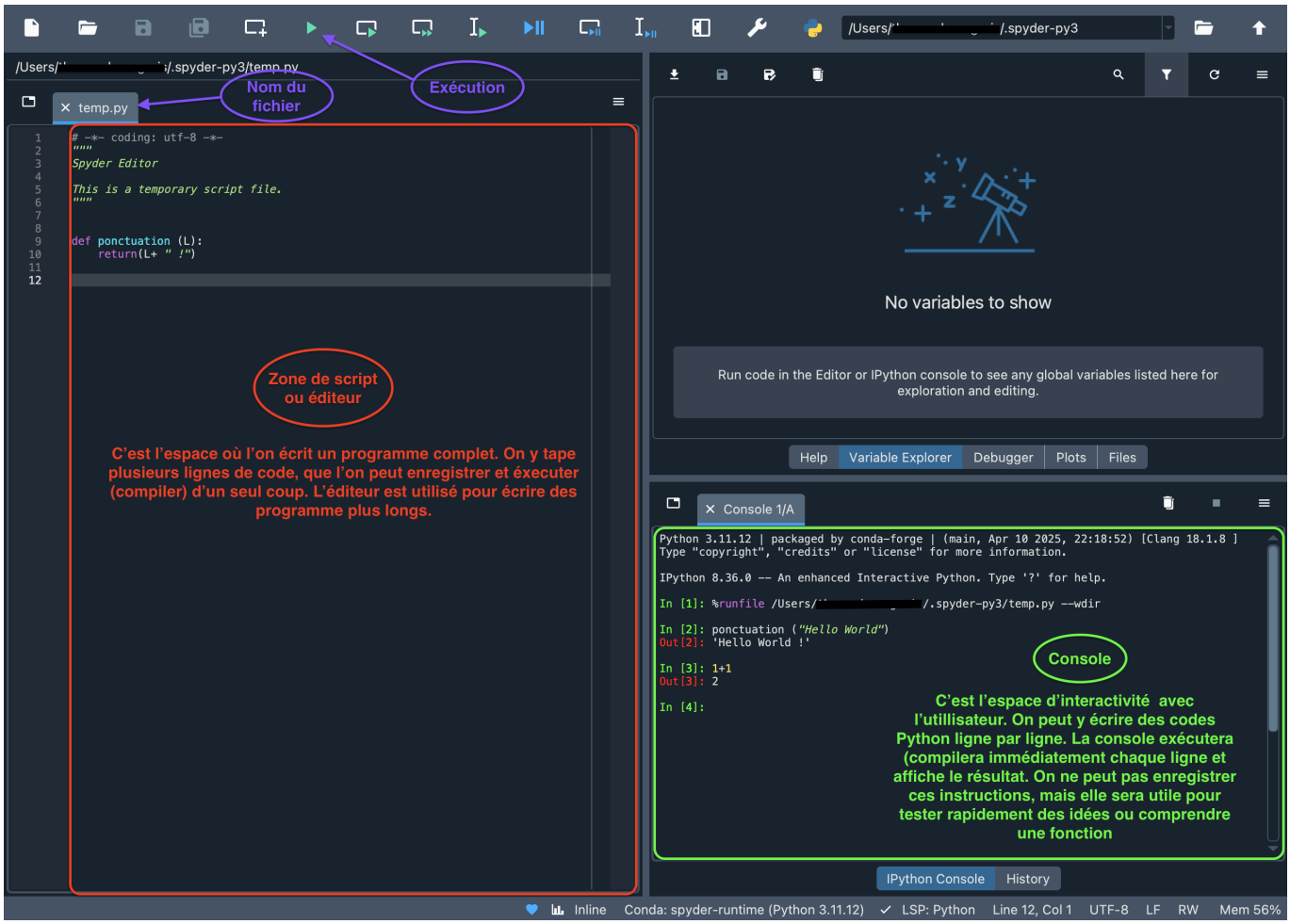
Partie II Python

Python est un langage informatique interprété, c'est-à-dire que l'utilisateur (vous) écrit un programme Python, constitués d'un langage spécifique et de différentes instructions. Lorsque l'utilisateur exécute le programme, les différentes instructions le constituant sont interprétées par le processeur de l'ordinateur et produisent un ou des résultats.
Par exemple, le script suivant permet d'afficher « Hello World » (c'est le script de départ de tout néophyte sur un langage de programmation) :
💻 :

```
1 x="Hello World"
2 print(x)
```

Comment coder en Python ? Sur un ordinateur. Et comment l'exécuter (on dira aussi le compiler) ? À l'aide d'un environnement de développement (en gros c'est une application qui nous permet de coder avec le langage Python), au lycée on utilisera : **Spyder**.

A - Interface



Nous avons deux zones principales :

- La **zone de script** ou **éditeur** ;
- La **console**.

Une fois compiler à l'aide du triangle ▶, l'application me précise s'il y a ou non des erreurs dans mon code et à quelle ligne se situe mon erreur. Elle me précise brièvement en anglais quelle est mon erreur. Si j'ai aucune erreur, généralement rien ne se passe car je ne lui ai rien demandé de faire précisément.
Ici dans l'exemple précédent, j'ai pu coder dans l'**éditeur** un programme permettant d'ajouter une ponctuation (un point d'exclamation) à la fin d'une chaîne de caractères.
💻 :

```
def ponctuation (L) :
        return( L+ " !")
```

La **console** se reconnaît à l'aide d'un enchaînement :

- textbfIn[...] (là où vous appeler des instructions à faire) **Out[...]** (les résultats renvoyés par la machine) ;
- De triple chevron >>>

💻 :

```
>>> ponctuation ("Hello World")
'Hello World !'
>>> 1+1
2
```

Ici j'ai demandé deux choses à la console :

1. À l'aide de la fonction ponctuation, mettre un point d'exclamation à ma chaîne de caractère `Hello World` ;
2. Faire le calcul 1+1.

Elle m'affiche sous chaque ligne le résultat obtenu.

B - Langage

Pour coder en Python, il faut connaître le langage particulier qu'il utilise. Précédemment nous avons le pseudo-code qui est utile pour coder à l'écrit sur un papier, avant de coder quoi que ce soit on prendra l'habitude de réfléchir à notre code sur papier.

B.1 - Opérations sur les types numériques

Tester sur votre ordinateur les instructions suivantes et noter les résultats obtenus :
💻 :

```
>>>12+5
...
>>>7-2
...
>>>6*3
...
>>>4**2
...
>>>5/2
...
>>>5//2
...
>>>5%2
...
>>> round(2.678,2)
...
>>> abs(-4)
...
```

Après avoir exécuter chacune des instructions préciser à quelles opérations mathématiques correspondent chacune des opérations Python suivantes :

Mathématiques	$a+b$	$a\times b$	$\frac{a}{b}$	a^b	Division euclidienne de a par b		Arrondir a à 10^{-n} près	Valeur absolue de a
					le quotient	le reste		
Python	...	...	...	...	...	...	...	...

Information :

- Python respecte les priorités opératoires habituelles en mathématiques et le rôle des parenthèses dans un calcul.
- Comme sur une calculatrice, le point décimal remplace la virgule dans l'écriture d'un nombre rationnel.

Erreur fréquente : On fera bien attention à ne pas noter la virgule décimale mais bien le point décimale, Python ne reconnaîtra pas : `2,1`. Cela est une très grande source d'erreurs...

À retenir : Différence entre division et quotient ?
Le quotient $\frac{a}{b}$ noté par : `a/b` est systématiquement un nombre réel c'est-à-dire un nombre flottant alors que le quotient de la division eucldienne de a par b noté par : `a//b` sera toujours un nombre entier (on verra plus en détails cette différence plus tard). Par exemple :

```
>>>4/2
2.0
>>>4//2
2
```

Ainsi si vous avez l'assurance que b divise a, c'est-à-dire que $\frac{a}{b}$ est un nombre entier alors il vaut mieux privilégier le quotient de la division euclidienne plutôt que le quotient classique.

On peut se rendre compte que nous n'avons pas défini toutes les opérations, par exemple la racine carrée... Il va nous falloir enrichir le vocabulaire de notre langage Python pour cela.
Dans la vie, pour enrichir son vocabulaire on va dans une bibliothèque. En Python c'est pareil ! Le langage Python possède des bibliothèques pour enrichir ses possibilités, parmi les plus connus :

- **math** (celle qu'on utilisera le plus en Seconde) qui nous permet d'avoir la racine carrée, le nombre π, la partie entière...
- **numpy** qui facilite le calcul numérique et permet la gestion des tableaux de données et matrices (qui ne sont pas aux programmes de Seconde) ou alors nous permet d'avoir la plupart des fonctions usuelles...
- **scipy** qui est une alternative à numpy ;
- **matplotlib.pyplot** pour les graphiques.

Pour avoir accès à la racine carrée il faut ajouter au début du programme (et exécuter) l'instruction :

```
from math import sqrt
```

qui veut dire : aller chercher dans la bibliothèque « `math` » la signification de la fonction `sqrt`. On peut récupérer toutes les fonctions de la bibliothèque `math` en tapant :

```
from math import *
```

Information : Le langage Python utilise beaucoup de mots ou d'abréviations anglais, ici `sqrt` est pour *square root* qui signifie racine carrée.

Exemple :

- **Racine carrée :**

```
from math import *
>>> sqrt(42.25)
6.5
```

- π

```
from math import *
>>> 3*pi
9.42477796076938
```

Erreur fréquente : On fera bien attention de noter l'étoile multiplicative (*) pour une multiplication, Python ne reconnaitra pas : `3pi`. Ceci est également une grande source d'erreur.

★★☆☆☆ EXERCICE 10 (Opérations)

Compléter le tableau suivant et vérifier vos résultats en les tapant dans la console.

Opération	5 * 7	18 / 4	3 ** 2	8 // 5	7 % 3	sqrt(16)	2 * 7 + 1	2 * (7 + 1)	sqrt(11) ** 2
Résultat									

★★☆☆☆ EXERCICE 11 (Traduction)

1. Écrire en Python le calcul suivant : $\dfrac{1,2}{2+3,5}$ (on ne demande pas déterminer la valeur)

   .....

2. Une pièce rectangulaire a pour dimensions 5,45 m et 3,63 m. Quel calcul effectuer en Python pour obtenir son aire en m^2, arrondie à 10^{-2} près.

   .....

B.2 - Affectation

Pour affecter une valeur à une variable, étant donné qu'il n'y pas → sur votre clavier, on utilisera le symbole « = ».

Exemple :

```
x=24
x=2*x
```

Ici on affecte à la variable x la valeur 24 puis on fini par affecter lui le double x.

Attention : Le symbole « = » n'a pas le même sens en informatique et en mathématiques, il ne signifie pas que l'on a une égalité ou une équation.
Ainsi, en informatique, l'instruction « 300 = a » n'a pas de sens. On ne peut pas remplacer la valeur de 300 par la valeur de a.
De même, en informatique l'instruction « $a = b$ » n'a pas le même sens que l'instruction « $b = a$ ».

- D'une part : c'est la variable a qui va prendre la valeur de la variable b. C'est la valeur de a qui est modifié et rien d'autre ;
- D'autre part : c'est la variable b qui va changer.

★★☆☆☆ Exercice 12 (Géométrie)

Dans chaque cas, donner le résultat obtenu et interpréter géométriquement chaque résultat.

1. Premier algorithme :

```
l=5
L=10
>>>L*l
...
```

.....

2. Second algorithme :

```
from math import pi
r=3
>>>pi*r**2
...
```

.....

Considérons l'algorithme suivant :

★★½☆☆ Exercice 13 (Calcul)

```
m=50
v=12
e=0.5*m*v**2
```

Quelle est la valeur de la variable e ?

.....

.....

★★★☆☆ Exercice 14 (À quoi ça sert ?)

Considérons l'algorithme suivant :

```
u=n%10
d=n//10
m=10*u+d
```

1. Recopier le programme ci-dessus.
2. Initialiser la variable n, avec un nombre du type integer tel que $10 \leq n \leq 99$.
   (a) Donner la valeur de la variable u après exécution des instructions. .....
   (b) Donner la valeur de la variable d après exécution des instructions. .....
   (c) Donner la valeur de la variable n après exécution des instructions. .....
3. Quel est le but de cet algorithme ?

.....

.....

★★☆☆☆ **EXERCICE 15 (Calcul)**

On considère le script suivant :

```
1 from math import sqrt
2 a=sqrt(10)-1
3 b=(sqrt(35)+1)/4
4 >>>a*b
```

Quel est le résultat affiché lorsque l'on tape les instructions suivantes ?

.....

.....

B.3 - Affichage

Pour afficher le contenu d'une variable `X` nous avons deux possibilités :

- L'instruction `print(X)` qui affiche la valeur contenue dans la variable `X` ;
- L'instruction `X` qui affiche la valeur entre guillemet.

Exemple :

- À l'aide de la commande `print` :

```
1 X="Hello World"
2 >>>print(X)
3 Hello World
```

- À l'aide du nom de la variable :

```
1 X="Hello World"
2 >>>X
3 'Hello World'
```

★★★☆☆ **EXERCICE 16 (Réduction)**

Un commerçant accorde une remise sur des articles. On souhaite connaître le montant de la remise en euros. Voici un algorithme, en pseudo-code, donnant la situation du problème.

Saisir le prix de départ A
Saisir le pourcentage de remise P
$R \leftarrow A \times \dfrac{P}{100}$
Afficher R

1. Dans ce groupement de question on suppose que $A = 56$ et $P = 30$
   (a) Écrire l'algorithme en Python.

   .....

   .....

   .....

   .....

   (b) Calculer la valeur de la variable R.

   .....

.....

(c) Donner une interprétation concrète du résultat précédent.

.....

.....

2. Même question avec $A = 13$ et $P = 45$

.....

.....

.....

.....

.....

3. Compléter votre algorithme pour afficher le prix à payer B.

.....

.....

.....

.....

.....

.....

4. (a) Calculer la valeur des variables R et B lorsque $A = 159$ et $P = 24$.

.....

.....

(b) Donner une interprétation concrète des résultats précédents.

.....

.....

C - La fonction type

Dans un ordinateur, les nombres sont représentés par une écriture (binaire) qui n'est pas l'écriture décimale. De ce fait, on distingue en Python :

- Les **entiers** qui sont représentés de façon exacte, ils sont du type **int** ;
- Les autres nombres qui sont représentés de façon approché : on dit que ce sont des **flottants** ils sont du type float.

On peut se demander pourquoi il est intéressant de connaître le type d'un nombre ?
Pour savoir interpréter certains résultats :

- Les calculs sur les entiers sont exacts, même pour de très grands entiers, contrairement à une calculatrice qui donnerait une valeur approchée.

```
3**37
>>>450283905890997363
```

- En revanche, les calculs avec des non entiers (des flottants) donnent des résultats approchés même pour des calculs qui paraissent simples (en écriture décimale, qui n'est pas utilisée par l'ordinateur)...

```
0.1+0.7
>>>0.7999999999999999
```

Information : Les résultats d'opérations en Python

- Tout calcul comportant au moins un flottant donne un résultat flottant ;
- La somme, la différence, le produit de deux entiers, une puissance d'un entier d'exposant positif donnent des entiers ;
- Une division et une racine carrée donnent toujours un flottant (contrairement aux mathématiques qui par exemple : $\frac{12}{3} = 4$ un entier, mais en Python 12/3 donne 3.0 un flottant)

À retenir : On retiendra que toute opération assurant, pour tout nombre, un résultat entier nous donnera un résultat entier en Python. Mais dès lors qu'il existe des nombres pour lesquels le résultat ne sera pas entier, les inventeurs ont codé cette opération Python comme une opération en flottant.

Définition 6 : Type

La fonction `type` renvoie le type (la nature) d'une variable.

Exemple :

```
x=2
y=2.0
z="2"
type(x)
>>>
type(y)
>>>
type(z)
>>>
```

★★½☆☆ EXERCICE 17 (Type)

Compléter le tableau suivant.

Nombre	Résultat	Type
2 + 4		
2.5 + 3		
2.3 + 3.7		
3 ** 4		
3.5 * 2		
5 * 6		
18/3		
18//3		
18%3		

Comme on vient de voir il est parfois plus utile d'avoir une variable en type entier plutôt qu'en type flottant pour avoir des calculs exacts.
C'est pourquoi il est intéressant de savoir changer le type d'une variable Python.
Si l'on cherche à transformer une variable `X` en type : `type2` (qui peut être : int, float, str...) on utilise la commande : `type2(X)`. Ainsi la variable `X` a comme nouveau type `type2`.

Exemple :

Si on définit une variable `X` contenant 2.0 alors cette variable est du type flottant et que l'on souhaite la changer en type entier, on a juste à faire :

```
X=2.0
type(X)
>>> float
X=int(X)
type(X)
>>> int
```

On a affecté à la variable `X` le contenu de cette même variable transformé en entier.

Erreur fréquente : Tout n'est pas accepté!

Il faut être vigilant, on ne peut pas tout transformer en tout ce que l'on veut. Par exemple si une variable de type chaîne de caractère contient : "Hello World" vous ne pourrez pas la transformer en une variable de type entière. En effet votre logiciel compilation ne va savoir comment transformer une phrase en un entier...

★★½☆☆ EXERCICE 18 (Erreurs)

Un programmeur s'est trompé lorsqu'il a initialisé deux variables qui vont lui permettre de calculer l'aire d'un rectangle.

```
L="5"
l="2.5"
aire=L*l
>>>error
```

Corriger le code précédent, en ajoutant des lignes pour obtenir le bon résultat.

.....

.....

.....

.....

.....

D - Chaînes de caractères

Pour les chaînes de caractères, deux opérations sont possibles, l'addition et la multiplication :

Définition 7 : Opérations

- L'opérateur **addition** « + » concatène (met bout à bout) deux chaînes de caractères.
- L'opérateur **multiplication** « ⋆ » entre un nombre entier naturel et une chaîne de caractères duplique (répète) plusieurs fois une chaîne de caractères.

Exemple :

```
chaine="Hello"
print(chaine+"World")
>>> HelloWorld
print(chaine*3)
>>> HelloHelloHello
```

Attention :

- On fera bien attention de multiplier une chaîne de caractère par un nombre entier naturel. Si c'est un entier négatif il est clair que cela n'a aucun sens, de même si ce n'est pas un entier...

- On observe bien que les opérateurs + et ⋆ se comportent différemment suivant s'il s'agit d'une variable numérique d'une chaîne de caractères.
  2+2 est une addition qui renverra : 4 alors que "2"+"2" est un concaténation qui renverra "22".

Nous avons d'autres outils pour manipuler les chaînes de caractères :

Définition 8 :

- La fonction `len` donne la longueur d'une chaîne de caractère.
- Pour une variable `L` contenant une chaîne de caractère, la commande `L[i]` donne le `i+1`ème caractère de la chaîne `L`.

Exemple :

```
len("Bonjour")
>>> 7
len("Bonjour ")
>>> 8
```

On fera bien attention un espace est bien comptabiliser comme un caractère.

```
L="Bonjour"
L[0]
>>> 'B'
L[3]
>>> 'j'
```

★★★★☆ Exercice 19 (Prédire le (potentiel) résultat)

Donner le résultat affiché pour chacune des commandes suivantes :

```
(1+2)**3
>>>
"Da"*3
>>>
"Da"+3
>>>
("Pa"+"La")*3
>>>
("Da"*4)/2
>>>
str(4)*int("3")
>>>
int("3")+float("3.2")
>>>
str(3)*float("3.2")
>>>
str(3/4)*2
>>>
```

★★☆☆☆ Exercice 20 (Affichage)

Donner le résultat affiché pour chacune des commandes suivantes :

```
mot="Hello World"
```

```
len(mot)
>>>
mot[1]
>>>
mot[5]
>>>
```

★★½☆☆ **EXERCICE 21 (Somme et produit)**

Tester sur votre ordinateur le code suivant :

```
a="12"
b="3"
print(a+b)
print(a*b)
```

1. Expliquer pourquoi il affiche des résultats inattendus.

   .....

   .....

2. Corriger le code pour obtenir des résultats mathématiquement corrects.

   .....

   .....

   .....

   .....

   .....

   .....

E - Commande `input`

Mise en situation : Que se passe-t-il lorsque vous souhaitez déverrouiller votre téléphone à l'aide de votre code?

- Votre téléphone vous affiche une zone d'entrée de code et attend que vous saisissiez votre code;
- Votre téléphone traite ensuite cette information (vérifie si le code est correcteur ou non et fait ensuite des actions adaptées à la situation)

Cette situation est mise en œuvre par la commande `input`.

Exemple :

La commande suivante :

```
code=input("Quel est votre code ?")
>>> Quel est votre code ? ...
```

Ici cette commande va afficher la chaîne de caractère souhaité : "Quel est votre code?" ainsi qu'une zone de traitement des données qui est représenté ici par les ...
Après avoir complété la zone de traitement des données, nous avons affecté cette valeur à la variable `code`
Si mon code est 0000 j'ai alors :

```
code=input("Quel est votre code ?")
>>> Quel est votre code ? 0000
print(code)
```

```
>>>0000
```

Attention : La commande `input` affecte automatiquement des chaînes de caractères, il faudra faire attention avec la manipulation des variables ayant été initialisé à l'aide d'un `input`.

★★☆☆☆ Exercice 22 (Aire)

Écrire un programme Python qui permet de calculer l'aire d'un rectangle après avoir demandé la longueur et la largeur du rectangle de l'utilisateur.

.....

.....

.....

.....

.....

★★★★☆ Exercice 23 (Magie)

Un magicien demande à un spectateur :

- De penser à un nombre entier ;
- De le multiplier par 5 ;
- D'ajouter 7 au résultat obtenu ;
- De multiplier par 4 le résultat ;
- D'ajouter 6 au résultat ;
- De multiplier par 5 le résultat obtenu ;
- D'annoncer le résultat obtenu.

1. Le spectateur pense au nombre 4, quel nombre annonce-t-il ?

   .....

   .....

2. Écrire un programme Python qui demande un nombre entier, effectue les opérations demandées par le magicien et affiche le résultat final(celui que le spectateur annonce).

   .....

   .....

   .....

   .....

   .....

   .....

   .....

   .....

3. Le magicien trouve à chaque fois le nombre choisi au départ par le spectateur ! Soit il est très fort en calcul mental, si il a une astuce de magicien... Expliquer comment le magicien fait pour trouver le nombre choisi au départ.

.....

.....

.....

.....

.....

.....

.....

.....

★★★☆☆ **EXERCICE 24 (Équation)**

1. Écrire un programme Python qui résout les équation du type : $ax + b = 0$ où $a \neq 0$. Le tester sur votre ordinateur.

.....

.....

.....

.....

.....

.....

.....

.....

2. Écrire un programme Python qui résout les équations du type : $ax + b = cx + d$ où $a \neq c$. Le tester sur votre ordinateur.

.....

.....

.....

.....

.....

.....

.....

.....