

ACTIVITÉ 1: DÉCOUVERTE DU LANGAGE PYTHON

Les ordinateurs sont capables d'effectuer rapidement de nombreuses tâches, mais ils ont besoin pour cela d'instructions précises et ordonnées. Pour résoudre un problème, on peut décrire une suite d'instructions sous la forme d'un algorithme, puis la traduire dans un langage compréhensible par l'ordinateur. Dans cette activité, nous allons découvrir les principes de l'algorithmique et apprendre à écrire nos premiers programmes en langage Python.

Problématique: Comment traduire une suite d'instructions en un programme informatique ?

Objectifs: À la fin de l'activité, vous serez capable de :

- lire et comprendre un algorithme simple ;
- suivre l'évolution de variables ;
- utiliser des instructions simples : affectation, entrée, affichage, condition ;
- traduire un algorithme simple en Python.

PARTIE 1: ALGORITHMIQUE: DÉFINITION ET INSTRUCTIONS ÉLÉMENTAIRES

Définition:

- Un algorithme est une suite finie d'instructions et d'opérations permettant de résoudre un problème posé .
- Un algorithme peut par exemple s'écrire :
 - en **langage naturel**
 - dans un **langage de programmation** (Python, JavaScript, PHP, C++, ...)

Un exemple d'algorithme écrit en langage naturel:

Alice souhaite connaître le prix à payer pour l'achat de plusieurs articles identiques.

Compléter l'algorithme suivant afin de calculer et afficher le prix total à payer par Alice et un commentaire associé.

```
demander(prix)
demander(quantité)
total ← prix × quantité          ← signifie "prend la valeur"
afficher(total)
si total < 50 alors :
    afficher("Le prix total est inférieure à 50 euros")
sinon :
    afficher("Le prix total est supérieur ou égal à 50 euros")
```

Alice lance cet algorithme avec le **prix** valant 12 € et la **quantité** valant 5.

- Quelle sera la valeur finale de la variable **total** ?

..... Total = 60 (5 x 12 = 60)

- Quelle phrase va s'afficher ?

..... Comme Total = 60, Total est supérieur ou égal à 50 donc la phrase affichée sera :

..... "Le prix total est supérieur ou égal à 50 euros"

Vocabulaire à mémoriser:

AFFECTATION

L'instruction d'**affectation** permet d'attribuer une valeur à une variable.

Lorsque l'on veut donner une valeur à la variable x , on écrit alors $x \leftarrow \dots$ (on lit « x prend la valeur $\dots$ »).

VARIABLES

Les instructions dans un algorithme font intervenir des quantités, qui sont appelées **variables**. Elles sont repérées par un nom et possèdent une valeur qui peut varier.

On distinguera cette année **trois types de variables** :

- **entier** (*int*, comme integer en anglais, correspondant aux nombres entiers)
- **flottant** (*float*, correspondant aux nombres décimaux)
- **booléen** (*bool*, d'après le mathématicien George Boole, correspondant aux deux états Vrai ou Faux)
- **chaîne de caractères** (*string*, correspondant aux suites de caractères (texte), par exemple "bonjour")

PARTIE 2: LECTURE D'UN ALGORITHME A L'AIDE D'UN TABLEAU DE VARIABLE

Exercice 1:

Alice a écrit un algorithme. Pour chaque ligne, donner le type de la variable utilisée.

INSTRUCTIONS	variable	type
$a \leftarrow 10$	a	entier (<i>int</i>)
$b \leftarrow 3.1425$	b	décimal (<i>float</i>)
$c \leftarrow a + b$	c	$a + b = 13,1425$ décimal (<i>float</i>)
$d \leftarrow \text{"bonjour"}$	d	chaîne de caractères (<i>string</i>)
$e \leftarrow \text{Vrai}$	e	booléen (<i>bool</i>)

Exercice 2:

Bob a écrit l'algorithme suivant. Compléter le tableau des variables afin de suivre l'évolution des valeurs de a , b et c

INSTRUCTIONS	a	b	c
$a \leftarrow 10$	10	/	/
$b \leftarrow 2 \times a$	10	20	/
$c \leftarrow a + b$	10	20	30
$a \leftarrow c - 3$	27	20	30
$c \leftarrow c + 1$	27	20	31
afficher (c)	31		

Exercice 3:

Charlie a écrit l'algorithme suivant. Compléter le tableau des variables afin de suivre l'évolution des valeurs de a , b et c

INSTRUCTIONS	a	b	c
$b \leftarrow 10$	/	10	/
$a \leftarrow 2 + b$	12	10	/
si $a < 11$ alors :	Faux		
$b \leftarrow b + 1$			
$c \leftarrow b + 20$			
sinon :	Vrai		
$b \leftarrow b - 1$	12	9	/
$c \leftarrow b + 10$	12	9	19
afficher(c)	19		

Exercice 4:

Dave souhaite écrire un algorithme qui demande de saisir l'âge d'un élève et écrive suivant le cas :
 "tu es majeur(e)" ou "tu es encore mineur(e)"
 Compléter son algorithme ci dessous :

```

demander (age)
si ... age < 18 ... :
    afficher (... tu es encore mineur(e) ...)
sinon :
    afficher (... tu es majeur(e) ...)
    
```

PARTIE 3: PREMIÈRE DÉCOUVERTE DU LANGAGE PYTHON

Remarques importantes :

- Le langage Python est le langage de programmation le plus utilisé au monde.
- C'est un langage libre et multiplateformes qui sera utilisé au lycée en mathématiques, en sciences et en SNT.

Exemple:

Voici la traduction en python de l'algorithme de la partie 1.

```

1 prix= float(input("donne-moi le prix : "))
2 quantité = int(input("donne-moi la quantité : "))
3 total = prix * quantité
4 if total < 50 :
5     print("Le prix total est inférieur à 50 euros")
6 else:
7     print("Le prix total est supérieur ou égal à 50 euros")
    
```

Explications de ce programme :

- Ligne 1 :** input signifie "demander" et float signifie "convertir en décimal"
- Ligne 2 :** input signifie "demander" et int signifie "convertir en entier"
- Ligne 4 :** if signifie "si"
- Ligne 5 :** print signifie "afficher"
- Ligne 6 :** else signifie "sinon"

Application:

En s'inspirant du programme précédent, donner le code python de l'exercice 4 de la partie 2

```
age = float(input("Quel est ton âge ? : "))
```

```
if age < 18 :  
    print("tu es encore mineur(e).")
```

```
else :  
    print("tu es majeur(e).")
```

Exemple de travail sur un premier programme Python:

On considère l'algorithme suivant :

```
demander(a)          avec a un entier  
b ← a + 3  
b ← b × 2  
b ← b - 5  
afficher(b)
```

On exécute ce programme avec 14 pour valeur de a .

1. Traduire de cet algorithme en langage Python :

```
a = int(input("donne moi un entier"))  
b = a + 3  
b = b * 2  
b = b - 5  
print(b)
```

Compléter le tableau des variables suivant :

INSTRUCTIONS	a	b
demander(a) avec a un entier	14	/
$b \leftarrow a + 3$	14	17
$b \leftarrow b \times 2$	14	34
$b \leftarrow b - 5$	14	29
afficher(b)	14	29

BILAN DE L'ACTIVITÉ 1.1: Langage naturel et langage Python

Un algorithme peut être décrit en langage naturel, avec des phrases compréhensibles par tous. Pour être exécuté par un ordinateur, il doit être traduit dans un langage de programmation respectant des règles précises. Python est l'un de ces langages : chaque instruction possède une syntaxe particulière qu'il faut respecter. Un même algorithme peut donc être exprimé en langage naturel puis traduit en langage Python.

En Python :

En langage naturel :

Affectation :

- L'affectation se code avec `=`
`x = 10`
`y = x+10`

- L'affectation se code avec `←`
`x ← 10`
`y ← x + 10`

Entrées :

- On utilise l'instruction **input**
`valeur = input("Entrer une valeur :")`
`n = int(input("Entrer un entier n : "))`
`x = float(input("Entrer un décimal x : "))`

- On utilise l'instruction **demander**
`demander(valeur)`
`demander(n)` avec *n* un entier
`demander(x)` avec *x* un décimal

- L'instruction **int** permet de convertir une valeur en entier
- L'instruction **float** permet de convertir une valeur en décimal

Affichages :

- On utilise l'instruction **print**
`print(variable)`
`print("Bonjour")`
`print("Salut" , variable, "Ça va ?")`

- On utilise l'instruction **afficher**
`afficher(variable)`
`afficher("Bonjour")`
`afficher("Salut" , variable, "Ça va ?")`

- Pour afficher du texte on utilise les guillemets.
- Pour afficher plusieurs variables, on les sépare par des virgules.