

1 Qu'est-ce qu'un algorithme?

Définition 1 : Un **algorithme** est une suite finie d'actions élémentaires assemblées de manière logique.

Le but d'un algorithme est de réaliser une action complexe en la décomposant en actions simples, c'est-à-dire pouvant être effectuées sans réflexion.

Exemple : Considérons les actions simples suivantes :

- Action 1 : Poser $a = 1$.
- Action 2 : Effectuer le calcul $\frac{a}{2} + \frac{1}{a}$
- Action 3 : Noter a le résultat.

Effectuons maintenant ces actions simples dans l'ordre suivant : Action 1 puis Action 2 puis Action 3 puis Action 2 puis Action 3.

Cela donne : -
 -
 -
 -
 -

On obtient une approximation de $\sqrt{2}$

Pour obtenir une meilleure précision, on poursuit en renouvelant autant qu'on le souhaite les Actions 2 et 3, comme par exemple : Action 1 puis Faire quatre fois de suite Action 2 puis Action 3.

On obtient alors au final $a = \dots\dots\dots$

Ce procédé est un algorithme connu appelé aussi algorithme de Héron.

Les premiers algorithmes connus remontent à l'époque babylonienne. Un algorithme peut être exécuté « à la main », c'est-à-dire en effectuant soi-même une à une les actions élémentaires, mais il existe depuis les années 1970 un instrument de calcul permettant de réaliser bien plus rapidement l'ensemble de toutes ces actions : l'ordinateur.

Il faut alors traduire l'algorithme dans un langage de programmation, on dit que l'on **implémente** cet algorithme, la transcription est alors appelée implémente

Il existe différentes façons de présenter un algorithme. Notre but étant l'implémentation dans un langage de programmation, nous choisirons d'écrire les algorithmes en **pseudo-code**.

Il s'agit de la présentation et de la syntaxe la plus proche des langages de programmation.

Nous utiliserons Python comme langage de programmation.

Plusieurs possibilités pour installer *Python*. Celle que je vous recommande est la suivante : Installer la distribution Anaconda sur votre ordinateur. Elle est téléchargeable à l'adresse

<https://www.anaconda.com/products/individual>

Python sera directement installé sur votre ordinateur et vous aurez directement accès à un IDE (environnement de travail) pour coder en Python. L'IDE que nous utiliserons en classe (disponible également sur les ordinateurs du lycée) est *Spyder*.

2 Découverte de la console et des commandes de bases

Lors du lancement de Spyder, une fenêtre s'ouvre.

- En bas à droite de cette fenêtre se trouve la console. Elle permet de faire des calculs ou de définir de petites fonctions... Dans la console, on voit `In[1]`
- La fenêtre comprend à gauche un éditeur de texte dans lequel on peut écrire des scripts (ou programmes) un peu plus longs. On peut ainsi les enregistrer pour les utiliser lors d'une nouvelle session et/ou les exécuter. Attention : Lors de l'exécution, les messages et affichages éventuels apparaissent dans la console.

2.1 Calculs sur les nombres réels

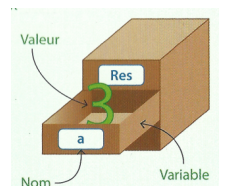
Évaluer les calculs suivants et en donner une interprétation :

1. $2 + 4 =$
2. $2 - 4 =$
3. $2 * 4 =$
4. $2 / 4 =$
5. $3 * * 2 =$
6. Appuyer alors plusieurs fois sur la touche $\uparrow$ puis sur la touche $\downarrow$. À quoi servent ces touches?
.....
.....

3 Les variables

Pour pouvoir réutiliser une donnée dans un programme informatique, il faut la stocker provisoirement dans un espace mémoire de l'ordinateur appelé **variable**.

Cet espace peut être considéré comme une boîte portant un nom et dans laquelle on range une donnée. Lorsqu'on stocke une donnée dans une variable, on dit que la variable est affectée à cette donnée.



L'affectation est une opération permettant de modifier la valeur d'une variable. La syntaxe de l'affectation est la suivante :

• en pseudo code : $\leftarrow$

• en python : `=`

Exemples d'affectation	Python
Affecter 'toto' à la variable mot	
Affecter 8 à la variable a	

Exercice 1 : Quelles sont les valeurs des variables après l'exécution des instructions suivantes ?

	A	B	C	D
A ← 1				
B ← 2				
C ← 3				
D ← A				
A ← C + 3				
B ← D * 5				
C ← 2 * C + A				
D ← A * C				

Convention le nom d'une variable doit respecter un certain nombre de règles :

- son premier caractère doit être une lettre (majuscule ou minuscule) ou « _ », les caractères suivants peuvent être des lettres, des chiffres, des symboles
- il ne doit pas être un mot réservé du langage (comme « print » ou « while »)
- par exemple : « aZe_Y44 », « _br_19 », « 4ahTR », « #qid4 ».
- pour faciliter la lecture d'un algorithme, il est essentiel de donner à chaque variable un nom en lien avec sa valeur. Par exemple, il est préférable qu'une variable ayant pour valeur 'Paul' s'appelle prenom. prénom plutôt que « d4j_5 ». Il vaut mieux cependant éviter les accents pas forcément bien reconnus sous Python.

4 Éditeur de texte

Taper directement dans la console a deux inconvénients : l'enregistrement n'est pas possible, et si plusieurs lignes d'instructions ont été tapées, les modifications ne sont pas aisées. Pour enchaîner plusieurs instructions, l'éditeur est l'outil approprié. On tape dans l'éditeur comme dans n'importe quel traitement de texte. L'éditeur reconnaît le langage Python : il met en couleur les mots-clef, un décalage de début de ligne appelé indentation se fait automatiquement lorsqu'on commence une boucle ou un test, il ferme automatiquement les parenthèses ouvertes...

- Placez-vous dans la partie gauche de la fenêtre.
- Dans cette partie, taper une première ligne de commande a=5. Sur la ligne en dessous, taper la commande b=8.
- Pour exécuter votre programme, cliquer sur l'icône « Play » (triangle vert). Cette icône permet à la fois d'enregistrer votre fichier et de l'exécuter.
- Dans la console, évaluer a puis b. Quel est le contenu de ces variables?
.....
- Dans le menu Fichier, choisir Enregistrer sous afin de sauvegarder ce document dans le dossier TP_1. On nommera ce fichier monPremierProg. Quel est le format de fichier proposé?



Vous sauverez régulièrement votre travail au risque de tout perdre et n'avoir plus que ses yeux pour pleurer.