



3ème * Sciences Informatiques

ALGORITHMIQUE & PROGRAMMATION

Leçon 1

Les structures de données Et les structures simples

Abassi rim

Lycée Ibn rachik Ezzahra

2021 - 2022

1. L'algorithmique et la programmation

1.1 - L'algorithme

Un **algorithme** est une suite finie (enchaînement , liste) ordonné (et logique) d'opérations (d'instructions / actions) permettant de résoudre un problème.

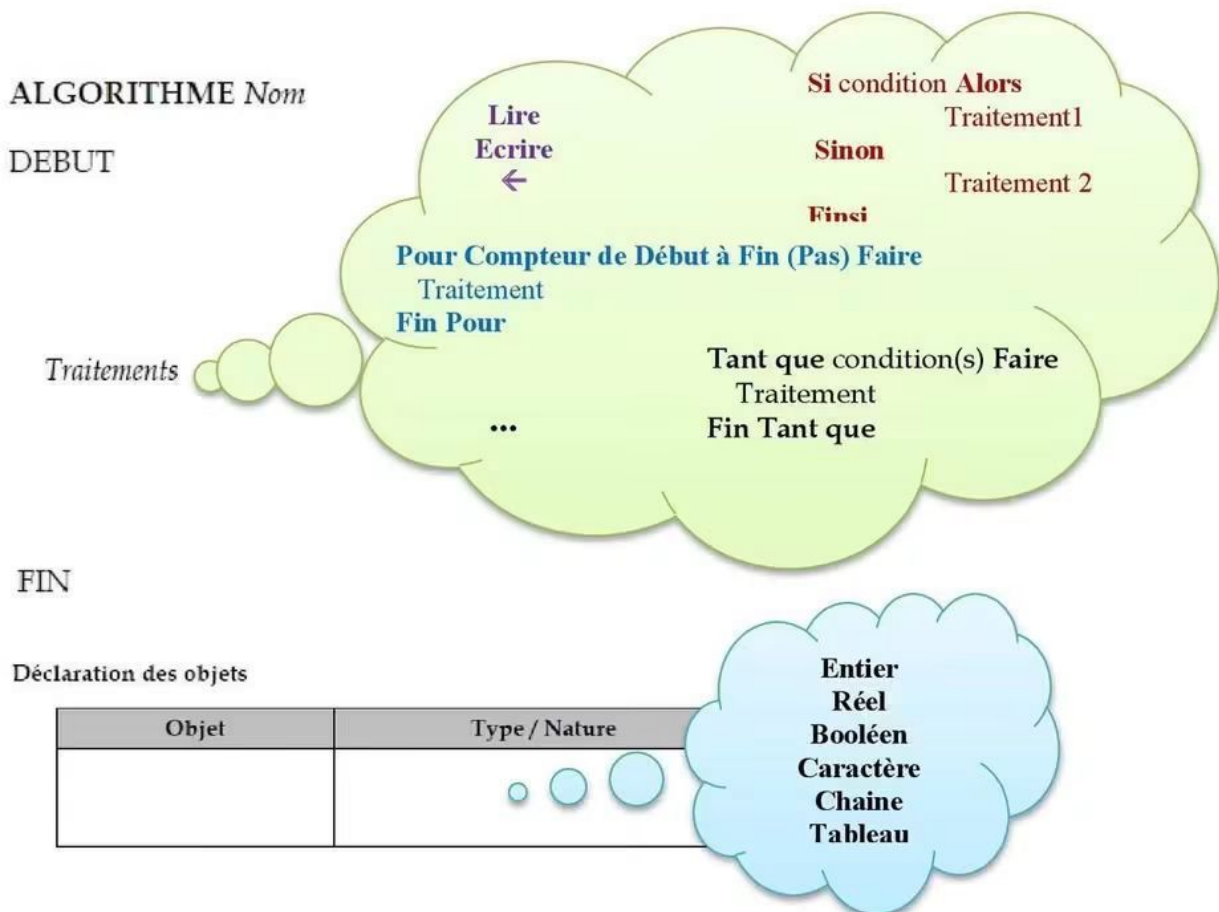
Autrement dit :

L'algorithme est la solution d'un problème informatique dans un langage naturel. Cette solution n'est pas compréhensible par l'ordinateur. Pour cela elle doit être traduite en un langage de programmation, ce qui donne un programme.

Un **algorithme** se compose de trois grandes parties :

- les informations dont on a besoin au départ ;
- la succession d'instructions à appliquer ;
- la réponse que l'on obtient à l'arrivée.

Pour écrire un algorithme il faut respecter la convention suivante :



1.2 - La programmation

la programmation, appelée aussi codage, est l'ensemble des activités qui permettent l'écriture des programmes informatiques en utilisant un langage de programmation.

Exemples de langage de programmation : Pascal, C++, Python

2. Les opérations élémentaires simples

Opération	En algorithmique
Entrée	Lire(objet)
Sortie	Ecrire("Message",Objet, Expression)
Affectation	Objet $\leftarrow$ Expression

3. Les structures de données

3.1 - Les variables

Une variable est « boîte » conçue pour contenir une information. Chaque variable doit avoir un nom pour qu'elle soit repérée par le programme. Il y a plusieurs types de variables :

- **numérique** : entier (int) , réel (float)
- **texte** : caractère, chaîne (string)
- **logique** : booléen (bool) ne peut prendre que deux valeurs : vrai ou faux

ATTENTION

- **En Python**, on n'a pas besoin de faire une déclaration des variables avant de les utiliser, le type d'une variable est déduit suivant la valeur qu'elle reçoit. il est parfois nécessaire de modifier en cours de route le type d'une variable. Pour déclarer une variable, rien de plus simple, il suffit de lui donner un nom et de lui affecter une valeur. Par exemple, je déclare la variable **x** et je lui affecte le nombre entier **13** $\rightarrow$ **x = 13**
- Exemples de déclarations
 - **x = int()** **#déclaration d'une variable de type entier**
 - **y = float()** **#déclaration d'une variable de type réel**
 - **z = bool()** **#déclaration d'une variable de type booléen**
- Le langage Python est sensible à la casse, c'est à dire que **x** est différente de **X**.
- Le symbole **#** est utilisé pour un commentaire en Python

3.2- Les opérateurs

Pour agir sur les variables, on peut utiliser des opérateurs qui dépendent du type de variables.

Il existe 3 types d'opérateurs :

Opérateurs arithmétiques

Opération	En Algorithmique	En Python
Addition	+	+
Soustraction	-	-
Multiplication	*	*
Division réelle	/	/
Division entière	div	//
Rester de la division entière	mod	%
Exponentiation	^	**

Opérateurs de comparaison

Opération	En Algorithmique	En Python
Egal	=	==
Différent	≠	!=
Strictement supérieur	>	>
Strictement inférieur	<	<
Supérieur ou égal	≥	>=
Inférieur ou égal	≤	<=
Appartient	∈	in
N'appartient pas	∉	not in

Opérateurs logiques

Opération	Opérateur
Négation	Non
Conjonction	Et
Disjonction	Ou
Disjonction exclusive	Ouex



ATTENTION

Il faut faire attention et bien distinguer l'instruction d'affectation = du symbole de comparaison ==.

3.3- Types de variables en Python

3.3.1 Les nombres

Il y a trois types numériques en Python :

- Le type entier **int** permet de représenter n'importe quel nombre entier, peu importe sa taille.
- Le type flottant **float** permet de représenter des nombres comportant une partie décimale (comme 3.14 ou $2.1e - 23$), compris entre 10^{-324} et 10^{308} .
- Le type complexe **complex** permet de représenter des nombres complexes, où le nombre imaginaire se note j (par exemple $3.1+5.2j$)

Les fonctions mathématiques & les nombres aléatoires

Algorithmique	Rôle
Arrondi (x)	Retourne l'entier le plus proche de la valeur de x.
RacineCarré (x)	Retourne la racine carrée d'un nombre x positif.
Aléa (vi, vf)	Retourne un entier aléatoire (au hasard) de l'intervalle [vi, vf].
Ent (x)	Retourne la partie entière de x.

3.3.2 Les booléens

Les variables de type booléen peuvent prendre deux valeurs : faux et vrai, en Python **False**, **True**.

3.3.3 Les caractères

Un caractère peut être un alphabet, un chiffre ou un symbole. Il est présenté entre guillemets (simples ou doubles). Les caractères sont ordonnés selon leurs codes ascii.

Algorithmique	Rôle	Exemple
Ord (c)	Retourne le code ascii du contenu de la variable c de type caractère	$X \leftarrow \text{ord}('A')$ $X = 65$
Chr (d)	Retourne le caractère dont le code ascii est d	$Y \leftarrow \text{chr}(65)$ $Y = 'A'$

3.3.4 Les chaînes de caractères

• Présentation

Une chaîne de caractères est une séquence de caractères entre guillemets (simples ou doubles).

En Python, les éléments d'une chaîne sont indexés à partir de 0 et non de 1.

-On peut accéder au $i^{\text{ème}}$ caractère d'une chaîne de caractères.

Exemple :

```
>>> s = 'Anis ELBAHI'  
>>> print(s[0])
```



Ce bloc affiche le caractère **A**



ATTENTION

Une chaîne de caractères est un objet **immuable**, c'est-à-dire ses caractères ne peuvent pas être modifiés par une affectation et sa longueur est fixe.

• Manipulation d'une chaîne

On peut extraire une sous-chaîne en déclarant l'indice de **début (inclus)** et l'indice de **fin (exclu)**, séparés par deux points : **s[i:j]**, ou encore une sous-chaîne en déclarant l'indice de début (inclus), l'indice de fin (exclu) et le pas, séparés par des deux-points : **s[i:j:k]**.

Algorithmique	Rôle
Long (ch)	Retourne le nombre de caractères de la chaîne ch .
Pos (ch1, ch2)	Retourne la première position de la chaîne ch1 dans la chaîne ch2 .
Convch (x)	Retourne la conversion d'un nombre x en une chaîne de caractères.
Estnum (ch)	Retourne Vrai si la chaîne ch est convertible en une valeur numérique, elle retourne Faux sinon.
Valeur (ch)	Retourne la conversion d'une chaîne ch en une valeur numérique, si c'est possible.
Sous_chaine (ch, d, f)	Retourne une partie de la chaîne ch à partir de la position d jusqu'à la position f (f exclue).
Effacer (ch, d, f)	Efface des caractères de la chaîne ch à partir de la position d jusqu'à la position f (f exclue).
Majus (ch)	Convertit la chaîne ch en majuscules.

Les fonctions sur les caractères et les chaînes

NB : On utilise l'opérateur + pour **concaténer** deux chaînes.

L'indice du premier élément d'une chaîne est 0.

Traduction en Python

<i>En algorithmique</i>	<i>En Python</i>
Types numériques	
Racine carré (a)	sqrt(a)
ent(a)	trunc (a)
Arrondi(a)	round(a,b)
Aléa(a,b)	randint(a,b)
Type caractère	
ord (c)	ord (c)
chr (x)	chr (x)
Type chaîne de caractères	
Long (ch)	len (ch)
Pos (ch1 , ch2)	ch2.find (ch1)
Convch (x)	str (d)
Estnum (ch)	ch.isdigit() ou ch.isnumeric()
Valeur (ch)	int (ch) ou float (ch)
Sous-chaîne (ch , d , f)	ch [d : f]
Effacer (ch , d , f)	ch = ch [: d] + ch [f :]
Majus (ch)	ch.upper ()

Affectation, Saisie et Affichage	
Syntaxe	Rôle
x=12.5	Affecter la valeur 12.5 dans la variable x
x=y	Affecter la valeur de la variable y dans la variable x
x,y=1.4, 3.65	Affecter simultanément la valeur 1.4 dans la variable x et la valeur 3.65 dans la variable y (x=1.4 et y =3.65)
n=int(input(" saisir n"))	Demande à l'utilisateur de saisir un entier, attendre la saisie et affecte la donnée dans la variable n.
n=float(input(" saisir n"))	Demande à l'utilisateur de saisir un réel, attendre la saisie et affecte la donnée dans la variable n.
ch=input(" saisir une chaîne")	Demande à l'utilisateur de saisir une chaîne, attendre la saisie et affecte la donnée dans la variable ch
n=eval(input(" saisir n"))	Demande à l'utilisateur de saisir une expression valide, attendre la saisie et affecte le résultat de cette évaluation dans la variable n.
Print(m)	Affiche dans la console la valeur de la variable m