

## Notion d'algorithme et les instructions de base

### 1-Notion d'algorithme

Un algorithme est une suite **d'actions** ou **d'instructions** qui doivent être exécutées dans un **ordre bien déterminé** pour résoudre un problème (ou réaliser un travail).

#### 1-1- Caractéristiques d'un algorithme

- La réalisation d'un algorithme est un acte créatif basé sur la logique
- Un algorithme doit être fini et doit se terminer après un nombre fini d'opérations.
- Un **même problème** peut être résolu au moyen de **plusieurs algorithmes**.

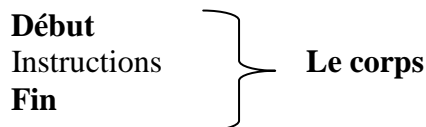
### 2-La représentation d'un algorithme

On peut représenter un algorithme à l'aide d'un **pseudo-code** ou d'un **organigramme**.  
Un algorithme écrit en **pseudo-code** est composé de trois parties suivantes :

**L'en-tête**, **la partie déclarative** et **le corps**

**L'en-tête**

**La partie déclarative**



#### A- L'en-tête d'un algorithme

**L'en-tête** est constitué du mot **Algorithme**, suivi d'un **nom** identifiant l'algorithme.

**Exemple**

- Algorithme** Somme ;
- Algorithme** Facture ;

#### B- La partie déclarative d'un algorithme

**La partie déclarative** comprend une liste des variables et des constantes utilisés et manipulés dans le corps de l'algorithme

#### 1- Les données d'un algorithme

Les données sont des informations nécessaires au déroulement d'un algorithme. On distingue deux catégories : les **constantes** et les **variables**.

##### 1-1- Les constantes

Une constante est une donnée fixe qui **ne varie pas** durant l'exécution d'un algorithme.  
Une constante est caractérisée par son **nom** et sa **valeur (fixe)**

**Syntaxe :**

Constante Nom \_Constante = valeur ;

**Exemple :**

Constante **Pi** =3,14 ;  
Constante **B** = 6 ;  
Constante **Mois** = 'Mars' ;

## 1-2- Les variables

Une variable est un objet dont le contenu peut être modifié par une **action** durant l'exécution d'un algorithme.

Une variable est caractérisée par son **nom**, sa **valeur** et son **type**.

### Syntaxe :

**Variable** Nom\_Variable : **Type** ;

## 1-3- Les types de données

Type de données	Numérique		Alphanumérique		Booléen
	Entier (sans la virgule)	Réel (Avec et sans la virgule)	Caractères	Chaîne de caractères	
<b>Exemples</b>	-345 178 2012	-123,56 56,12 3	$4,1 \times 10^{38}$ 18 -123	'A,' '@' '2 ' '? ' '+' '\$'	'Ibn Batouta' '49', '3872' 'Bonjour'
					Vrai Faux

### Exemple

Variable **nb\_élèves** : entier ;

Variable **note** : réel ;

Variable **Prénom** : chaînes de caractères ;

Variable **A** : booléen ;

Variable **Opérateur** : caractère ;

### Remarque

- '32' est différent de 32 parce que **32** est trente deux par contre '32' représente la suite des chiffres 3et2.
- Une variable de type numérique ne peut pas recevoir une variable de type alphanumérique ou de type booléen.

### Exercice d'application

Rédiger l'en-tête et la **partie déclarative** d'un algorithme qui permet de calculer et afficher la surface d'un cercle.

## C- Le corps d'un algorithme

Le corps d'un d'algorithme est une suite d'instructions ou des tâches à exécuter.

### 1 - Les instructions de base

#### 1-1- Lire (Fonction d'entrée)

Instruction qui permet d'entrer des données **tapées** au **clavier**.

### Syntaxe :

**Lire** (variable1) ;  
**Lire** (variable2) ;  
**Lire** (variable1, variable2) ;

### Exemple :

**Lire** (note) ;

**Lire** (A, B) ;

#### 1-2- L'écriture (sortie)

Instruction qui permet **d'afficher** le contenu d'une variables ou/et un message sur l'**écran**

## Syntaxe :

**Ecrire** (variable) ;  
**Ecrire** ('message') ;  
**Ecrire** ('message', variable) ;

### Exemple :

Soit **A** est une variable.

**Ecrire** (A) : signifie affiché sur l'écran le **contenu** de la variable A.

**Ecrire** ('donnez votre nom : ') : signifie affiché sur l'écran le message suivant : **donnez votre nom :**

### 1-3- L'affectation

L'affectation est une **opération** qui consiste à attribuer une valeur à une variable. Elle est représentée par une flèche orientée à gauche ←

## Syntaxe :

Variable ← Valeur ou expression

### Exemple

**A** ← 2 : la variable A reçoit la valeur 2

**B** ← A+1 : la variable B reçoit le contenu de A plus 1

**Nom** ← 'Mohamed' : la variable Nom reçoit la valeur Mohamed

### Remarque :

- L'instruction d'affectation ne modifie que ce qui est situé à gauche de la flèche.
- La valeur de la partie droite doit obligatoirement être du type de la variable dont la valeur est modifiée.

## 2- Expressions et opérateurs

### 2-1- Expression

-Une **expression** est un ensemble de valeurs reliées par des opérateurs, et équivalent à une seule valeur

### Exemple :

4+2, 5\*6, 4<9, A>6, C+B,...

### 2-2- Opérateur

-Un opérateur est un signe qui relie deux valeurs, pour produire un résultat.

### Exemple :

+, -, \*, /, ^, <, et, non, ou, ...

Opérateurs numériques	Opérateurs alphanumériques	Opérateurs booléens et Comparaisons
<b>+</b> : addition <b>-</b> : soustraction <b>x</b> : multiplication <b>/</b> : division <b>^</b> : puissance <b>Mod</b> : Reste de la division entière <b>Div</b> : Division entière	<b>+</b>	<b>ET</b> <b>OU</b> <b>NON</b> <b>&gt;</b> , <b>=</b> , <b>&lt;</b> , <b>&gt;=</b> , <b>&lt;=</b> , <b>≠</b>

### 2-3- Table de vérité des opérateurs booléens

<i>A</i>	<i>B</i>	<i>A ET B</i>	<i>A OU B</i>	<i>NON A</i>
<b>Faux</b>	<b>Faux</b>	<b>Faux</b>	<b>Faux</b>	<b>Vrai</b>
<b>Faux</b>	<b>Vrai</b>	<b>Faux</b>	<b>Vrai</b>	<b>Vrai</b>
<b>Vrai</b>	<b>Faux</b>	<b>Faux</b>	<b>Vrai</b>	<b>Faux</b>
<b>Vrai</b>	<b>Vrai</b>	<b>Vrai</b>	<b>Vrai</b>	<b>Faux</b>

Exemples d'application

<b>A</b> ← $4^2$	<b>C</b> ← 'Lycée '	<b>F</b> ← $3 < 5$
<b>B</b> ← $8 \times 5$	<b>D</b> ← 'Ibn Batouta'	<b>G</b> ← $2 > 4$
<b>N</b> ← $A + B$	<b>E</b> ← $C + D$	<b>H</b> ← $1 < 8$
<b>P</b> ← $N - 20$		<b>K</b> ← (F et G) ou H
<b>R</b> ← $B / 3$		<b>J</b> ← G et F et H
<b>X</b> ← $B \text{ Mod } 3$		<b>M</b> ← Non G
<b>Y</b> ← $B \text{ Div } 3$		

Résultats

<b>A</b> vaut: .....	<b>E</b> vaut : .....	<b>F</b> prend la valeur: .....
<b>B</b> vaut: .....		<b>G</b> prend la valeur: .....
<b>N</b> vaut: .....		<b>H</b> prend la valeur: .....
<b>P</b> vaut: .....		<b>K</b> prend la valeur: .....
<b>R</b> vaut : .....		<b>J</b> prend la valeur: .....
<b>X</b> vaut: .....		<b>M</b> prend la valeur: .....
<b>Y</b> vaut: .....		

## 3- Concevoir un algorithme

Pour concevoir un algorithme, il faut suivre les étapes suivantes :

### Etape1

Lire et comprendre bien l'énoncé du problème à résoudre

### Etape 2

- Définir les résultats du problème (les sorties) ;
- Définir les données du problème (les entrées) ;
- Définir le traitement (les relations permettant d'obtenir les résultats à partir des données) ;

### Etape 3

Ecrire l'algorithme on respectant la structure pseudo-code

### Exemple1

Un algorithme qui demande la valeur du rayon pour calculer la surface d'un cercle

**Etape1** : on veut calculer la surface d'un cercle

**Etape 2 :**

**Résultat**

La surface du cercle      Surf

**Données**

Le rayon      R

3,14

Pi

**Traitement**

Surf = Pi \* R\*R

**Etape 3**

Algorithme **Surface\_cercle** ;

**L'en-tête**

Constante Pi = 3,14 ;

Variable R, Surf : Réel ;

**Les déclarations**

**Début**

**Ecrire** (' Donnez la valeur de rayon: ' ) ;

**Lire** (R) ;

**Le corps**

Surf ← Pi\*R^2 ;

**Ecrire** (' La surface de cercle est : ', Surf) ;

**Fin.**