



*« Un langage de programmation est une convention pour donner des ordres à un ordinateur. Ce n'est pas censé être obscur, bizarre et plein de pièges subtils. Ça, ce sont les caractéristiques de la magie. »*  
(Dave Small)

---

## Sommaire

Algorithmique . . . . .	2
Les variables . . . . .	2
Les différents types de variables . . . . .	3
Les tests . . . . .	3
La boucle « POUR » . . . . .	4
La boucle « TANT QUE » . . . . .	5

---

## Définition

Algorithmique

Le mot *algorithme* vient du mathématicien Al-Khwarizmi.

Il désigne une succession d'instructions compréhensibles par un programme donné qui permettent au final d'arriver à un résultat.

Construire un programme est donc comme expliquer à une personne le chemin qu'elle doit prendre pour aller d'un point A à un point B de sorte qu'elle ne se perde pas. Il faut donc que chaque étape soit simple.

L'algorithmique, l'ADN et les **Shadoks** ont une chose en commun : tout part de quatre termes.

- Les Shadoks ont un univers lexical composé des termes « Ga », « Bu », « Zo » et « Meu », ce qui ne leur a pas empêché de prononcer des formules qui valent le détour.
- L'ADN a pour base l'Adénine, la Guanine, la Cytosine et la Thymine, ce qui ne nous a pas empêché d'exister.

Vous le voyez, tout est une question d'agencement ...

## Propriété

Les instructions de base

Un algorithme est composé de quatre types d'instructions :

- l'affectation de variables
- la lecture / écriture
- les tests
- les boucles

## Définition

Les variables

Les *variables* d'un algorithme sont des lettres dans lesquelles seront stockées des informations (numériques ou littérales).

## Exemples

Les variables

Comme en algèbre, la variable *a* pourra prendre différentes valeurs numériques comme  $-2$ ,  $7$ , ...

La variable *temp* pourra prendre pour valeur une phrase.

On pourra affecter à une variable une valeur dès le début de l'algorithme, plus tard ou bien l'utilisateur pourra informer cette valeur à chaque exécution du programme.

Mais pour utiliser une variable, il faut impérativement la déclarer au début de l'algorithme (afin que celui-ci comprenne que c'est une variable).

## Remarque

Le nom d'une variable ne doit pas comporter de signes de ponctuation ni d'espaces.

## Exemple

*Exemple de déclaration de variables*

**DEBUT\_ALGORITHME**

**VARIABLES :** *A (entier), B (réel), RES (alphanumérique)*

**FIN\_ALGORITHME**

Ici, nous avons spécifié que nous allons utiliser trois variables : la première (appelée « *A* ») sera de **type numérique** et sera en plus un nombre entier, la seconde (appelée « *B* ») sera aussi de type numérique et sera un nombre réel, et une troisième (appelée « *RES* ») sera du **type alphanumérique**, c'est-à-dire composée de caractères et de chiffres.

## Définition

*Les différents types de variables*

*Il existe trois types de variables :*

- *Le type numérique (entier ou réel)*
- *Le type alphanumérique (une chaîne de caractères)*
- *Le type booléen (VRAI ou FAUX - TRUE ou FALSE - 1 ou 0)*

## Exemple

*Exemple d'algorithme simple*

**DEBUT\_ALGORITHME**

**VARIABLES :** *A (entier), B (entier)*

**LIRE** *A*

**LIRE** *B*

**ECRIRE** *A\*B*

**FIN\_ALGORITHME**

Cet algorithme demande à l'utilisateur de rentrer deux valeurs entières, puis affiche leur produit.

## Définition

*Les tests*

*Dans un algorithme, les tests se font sur les variables et se formulent de la façon :*

**SI** *<condition(s)>* **ALORS** *<instruction(s)>* **SINON** *<instruction(s)>*

## Exemple

*Exemple d'algorithme simple*

### DEBUT\_ALGORITHMME

**VARIABLES :** *A (entier), B (entier)*

**LIRE** *A*

**LIRE** *B*

**SI** *A > B* **ALORS**

**ECRIRE** *« A est plus grand que B »*

**SINON**

**ECRIRE** *« A est plus petit que B »*

**FIN DU SI**

### FIN\_ALGORITHMME

## Exemple

*Exemple d'algorithme à tests imbriqués*

### DEBUT\_ALGORITHMME

**VARIABLE :** *temp (entier)*

**ECRIRE** *« Entrez la température de l'eau : »*

**LIRE** *temp*

**SI** *temp ≤ 0* **ALORS**

**ECRIRE** *« C'est de la glace ! »*

**SINON**

**SI** *temp < 100* **ALORS**

**ECRIRE** *« C'est du liquide ! »*

**SINON**

**ECRIRE** *« C'est de la vapeur ! »*

**FIN DU SI**

**FIN DU SI**

### FIN\_ALGORITHMME

## Définition

*La boucle « POUR »*

*La boucle s'écrivant :*

**POUR** *<variable>* **DE** *<valeur 1>* **A** *<valeur 2>* **FAIRE** *<instruction(s)>*

*sert à exécuter une ou plusieurs instructions lorsque la variable prend les valeurs allant de « valeur 1 » à « valeur 2 ».*

## Exemple

### DEBUT\_ALGORITHMME

**VARIABLES :** *somme (entier), n (entier), i (entier)*

*somme* **PREND LA VALEUR** 0

**LIRE** *n*

**POUR** *i* **DE** 1 **A** *n* **FAIRE**

*somme* **PREND LA VALEUR** *somme+i*

**FIN DU POUR**

**ECRIRE** *somme*

### FIN\_ALGORITHMME

Cet algorithme demande à l'utilisateur de saisir une valeur entière  $n$ , puis calcule la somme des  $n$  premiers entiers.

## Définition

*La boucle « TANT QUE »*

*La boucle s'écrivant :*

**TANT QUE** *<condition(s)>* **FAIRE** *<instruction(s)>*

*sert à exécuter une ou plusieurs instructions tant que les conditions sont vraies.*

## Exemple

### DEBUT\_ALGORITHMME

**VARIABLES :** *somme (entier), n (entier), i (entier)*

*somme* **PREND LA VALEUR** 1

*i* **PREND LA VALEUR** 2

**LIRE** *n*

**TANT QUE**  $i \leq n$  **FAIRE**

*somme* **PREND LA VALEUR** *somme+i*

*i* **PREND LA VALEUR** *i+1*

**FIN DU TANT QUE**

**ECRIRE** *somme*

### FIN\_ALGORITHMME

Cet algorithme fait la même chose que le précédent.