

INSTITUT SABDARIFA

M1 Géomatique & Technologies numériques

Algorithmique et programmation

Chapitre 1 : Variables et instructions

M. Amadou MBAYE

E-mail : mbayeamadou@mbayeamadou.com

Site web : www.mbayeamadou.com

23 novembre 2026

Plan du chapitre

Notion de variable

Opérations sur les variables

Syntaxe générale d'un algorithme

Instructions élémentaires

- Affectation
- Entrées et sorties

Un algorithme du quotidien

Du réveil au départ pour le cours d'algorithmique

Les opérations élémentaires à réaliser entre le moment ou le reveil sonne et le moment ou l'on sort pour aller faire le cours d'algorithmique :

1. Le réveil sonne.
2. Je me réveille.
3. Je me lève.
4. Je prépare le café.
5. Je prends le petit-déjeuner
6. J'enleve mon pyjama
7. Je me douche
8. Je m'habille
9. Je porte mes chaussures.
10. J'ouvre la porte.
11. Je sors.
12. Je ferme la porte

Peut on intervertir les actions 7 et 6, ou 9 et 7 ou encore 10 et 12 ?

Un algorithme du quotidien

Et si j'ai un parapluie ?

Structure alternative ou conditionnelle

La suite des opérations doit me permettre de ne prendre mon parapluie que dans la condition ou il pleuvrait.

Pour ce cas, nous disposons d'une structure alternative permettant de faire un choix.

Un algorithme du quotidien

De même, tous les jours, je répéte ces même actions jusqu'à ce que je suis en vacance (le réveil ne sonnera plus..)

Structure répétitive ou itérative

Dans ce cas, nous disposons d'une structure itérative ou répétitive permettant de répeter ces gestes jusqu'à ce que le réveil ne sonne plus.

Définition

Algorithme

Il fait passer d'un état initial un état final de façon **déterministe**.
L'algorithme doit respecter les régles suivantes :

- Il est défini sans ambiguité
- Il prend fin après un nombre fini d'opérations
- Il doit etre effectif
- Il manipule des objets qui doivent etre définies de maniere trés précise.

Définition

Algorithme vs Programme

- Un **algorithme** est une suite finie d'instructions précises et ordonnées qui agissent sur des données pour résoudre un problème.
- Un **programme** traduit ces instructions dans un langage de programmation pour que l'ordinateur puisse les comprendre et les exécuter.

Définition

Algorithme vs Programme

Il fait passer d'un état initial un état final de façon **déterministe**.
L'algorithme doit respecter les régles suivantes :

- Il est défini sans ambiguité
- Il prend fin après un nombre fini d'opérations
- Il doit etre effectif
- Il manipule des objets qui doivent etre définies de maniere trés précise.

Une suite d'actions ordonnées en séquence qui porte les objets d'un univers fini.
Un programme traduit ces actio dans un langage de programmation pour qu'un ordinateur puisse les exécuter.

Notion d'objet

Objet et valeur

- Les données manipulées et les résultats produits sont appelés des **objets**.
- Pour utiliser un objet, il faut connaître les valeurs qu'il peut prendre.

Notion d'objet

Objet et valeur

- Les données manipulées et les résultats produits sont appelés des **objets**.
- Pour utiliser un objet, il faut connaître les valeurs qu'il peut prendre.
- Si cette valeur ne peut pas être modifiée, nous parlons de **constante**, sinon nous parlons de **variable**.

Notion d'objet

Objet et valeur

- La récupération des données d'un algorithme se fait en mémoire centrale, à partir du clavier ou d'un autre périphérique d'entrée.

Notion d'objet

Objet, valeur et mémoire

- La récupération des données d'un algorithme se fait en mémoire centrale, à partir du clavier ou d'un autre périphérique d'entrée.
- Le programme conserve ses données en mémoire centrale.

Notion d'objet

Objet, valeur et mémoire

- La récupération des données d'un algorithme se fait en mémoire centrale, à partir du clavier ou d'un autre périphérique d'entrée.
- Le programme conserve ses données en mémoire centrale.
- Le processeur les lit et les traite lorsqu'il exécute les instructions.

Objectif d'un algorithme

Données, traitement et résultats

Un algorithme traite une ou plusieurs données pour produire un résultat ou des resultats.

$$\text{Données} \longrightarrow \text{Traitement} \longrightarrow \text{Résultats}$$

Objectif d'un algorithme

Données, traitement et résultats

Un algorithme traite une ou plusieurs données pour produire un résultat ou des resultats.

$$\text{Données} \longrightarrow \text{Traitement} \longrightarrow \text{Résultats}$$

Le(s) résultat(s) d'un algorithme peut (peuvent) servir de donnée(s) à un autre algorithme.

Plan du chapitre

Notion de variable

Opérations sur les variables

Syntaxe générale d'un algorithme

Instructions élémentaires

- Affectation
- Entrées et sorties

Définition d'une variable

Contenant et contenu

On peut représenter une variable par une case mémoire identifiée par un nom. La case contient une valeur.
Le nom sert à repérer un emplacement donné dans la mémoire centrale.

- Une variable en tant qu'emplacement va être considérée comme un contenant.
- sa valeur est le contenu.

Remarque :

- Deux variables peuvent avoir la même valeur, mais une variable ne peut pas avoir plusieurs valeurs en même temps.
- La valeur d'une variable peut varier au cours du programme.

Caractéristiques d'une variable

Identificateur, type et valeur

- Identificateur : donne un nom à la variable. Dans la convention de pseudo-code du cours, il commence par une lettre et ne contient pas d'espace.
- Type : détermine l'ensemble des valeurs admises et les opérations possibles.
- Valeur : élément courant appartenant à cet ensemble.

Déclaration et initialisation

Décrire les données avant le traitement

Dans le pseudo-code du cours, la déclaration indique le nom et le type de chaque variable. Un langage qui exige une déclaration peut utiliser le type pour réserver la place nécessaire en mémoire. Tel n'est pas le cas avec Python.

```
rayon, perimetre : réel
nombre : entier
```

Déclaration et valeur initiale

Une déclaration précise le type d'une variable. Son initialisation consiste à lui donner une première valeur.

La valeur d'une constante est fixée dans sa déclaration.
`PI = 3.14` fixe l'approximation de π utilisée dans le calcul du périmètre.

Les types de base

Quatre catégories de valeurs

Type	Exemples
Caractère	`'B'`, `'?'`, `'@'`
Entier	-43, 52
Réel	-97.43, 652
Booléen	`VRAI`, `FAUX`

Plan du chapitre

Notion de variable

Opérations sur les variables

Syntaxe générale d'un algorithme

Instructions élémentaires

- Affectation
- Entrées et sorties

Opérations sur les réels

Opérations possibles

Type	**Exemples**	**Opération**	**Symbole**
Réel	$-97{,}43$ et 652	Addition	$+$
		Soustraction	$-$
		Multiplication	$*$
		Division	$/$
		Puissance	$\hat{}$
		Pourcentage	%
		Comparaisons	$<, \leq, >, \geq, =, \neq$

Remarque :

En Python, la puissance s'écrit $**$

Opérations sur les entiers

Opérations possibles

Type	Exemples	Opération	Symbole
Entier	-43 et 52	Addition	$+$
		Soustraction	$-$
		Multiplication	$*$
		Division entière	`div`
		Modulo	`mod`
		Comparaisons	$<, \leq, >, \geq, =, \neq$

Remarque :

En Python, on écrit `7 // 2` pour la division entière et `7 % 2` pour le reste.

Caractères et booléens

Comparaison selon le codage caractères

L'ordre du code ASCII est utilisé pour comparer les caractères. Par exemple, `Z` a le code 90 et `a` le code 97. On obtient donc `Z` $<$ `a` dans cet ordre.

NON, ET et OU

Opération	**Pseudo-code**	**Python**
Négation	`NON A`	`not A`
Conjonction	`A ET B`	`A and B`
Disjonction	`A OU B`	`A or B`

Plan du chapitre

Notion de variable

Opérations sur les variables

Syntaxe générale d'un algorithme

Instructions élémentaires

- Affectation
- Entrées et sorties

Composition d'un algorithme

En-tête et corps

- L'**en-tête** contient le nom de l'algorithme et les déclarations de constantes et de variables. Les constantes précèdent les variables dans la convention du cours.
- Le **corps** contient les instructions à exécuter. Les mots DEBUT et FIN marquent ses limites.
- Un algorithme commence par le mot ALGORITHME suivi de son nom (identifie de l'algorithme)

Syntaxe générale

Écriture en pseudo-code

```
ALGORITHME Nom_algorithme
CONSTANTES nom_constante = valeur
VARIABLES nom_variable : type
DEBUT
  Instruction 1
  Instruction 2
  ...
FIN
```

Plan du chapitre

Notion de variable

Opérations sur les variables

Syntaxe générale d'un algorithme

Instructions élémentaires

- Affectation
- Entrées et sorties

Instructions courantes

Échanges et calculs

Le programme manipule des variables, des constantes, des valeurs littérales et des expressions. Les messages affichés peuvent aussi être des chaînes littérales.

La saisie transfère une valeur du clavier vers une variable.

L'affectation mémorise une valeur.

L'affichage communique un résultat à l'utilisateur.

Plan du chapitre

Notion de variable

Opérations sur les variables

Syntaxe générale d'un algorithme

Instructions élémentaires

- Affectation
- Entrées et sorties

Affectation

Nom à gauche, expression à droite

Dans $x \leftarrow y$, la valeur de y est lue, puis affectée à x.

L'instruction $y \leftarrow x$ agit dans le sens inverse.

À droite de la flèche, on peut écrire une constante, une variable ou une expression.

Par exemple,

$x \leftarrow 5 + 97$

donne à x la valeur 102.

Affectation

Modifier une valeur

$a \leftarrow a + 3$

demande de lire l'ancienne valeur de a, de lui ajouter 3 et de mémoriser le résultat dans a.

Si a vaut 4 avant l'instruction, il vaut 7 après.

En mathématiques, l'égalité $a = a + 3$ est impossible.

Affectation

Le sens de l'affectation

- $x \leftarrow y$ donne à x la valeur de y.
- $y \leftarrow x$ donne à y la valeur de x.

En mathématiques, l'égalité est symétrique : $x = y$ équivaut à $y = x$.

À gauche de la flèche

Dans une affectation simple, la cible doit être une variable.

- $y + 5 \leftarrow b$ est une affectation invalide.
- $y + 5 = b$ est une équation mathématique.

Suivre les affectations

Valeurs successives de deux variables

Instruction	a	b
$a \leftarrow 3$	3	Indéterminée
$b \leftarrow a + 2$	3	5
$a \leftarrow b \times 2$	10	5
$b \leftarrow b + 1$	10	6

Affectation

Exemple

```
ALGORITHME Affectation_de_valeur
VARIABLES
    a, b : Entier

DEBUT
    a <- 3          // a vaut 3, b est indetermine
    b <- a + 2      // a vaut 3, b vaut 5
    a <- b * 2      // a vaut 10, b vaut 5
    b <- b + 1      // a vaut 10, b vaut 6
FIN
```

Affectation

Initialisation des variables

Initialiser une variable consiste à lui donner une valeur avant sa première utilisation.

On peut initialiser une variable par affectation, par exemple $a \leftarrow 0$, ou par saisie au clavier. On peut aussi la réaffecter au cours du programme.

Affectation

Exemple

```
ALGORITHME Initialisation_de_variables
VARIABLES
    a, b : Entier

DEBUT
    // Initialisation des variables
    a <- 0
    b <- -23

    // Autres instructions
FIN
```

Plan du chapitre

Notion de variable

Opérations sur les variables

Syntaxe générale d'un algorithme

Instructions élémentaires

- Affectation
- Entrées et sorties

ÉCRITURE

Généralités sur l'instruction de saisie

- La saisie au clavier permet à l'utilisateur de communiquer des données au programme.
- Le programme affecte la valeur saisie à la variable indiquée.
- L'exécution reprend lorsque l'utilisateur valide sa saisie.

Remarque

La saisie peut donner sa première valeur à une variable. Une initialisation préalable est donc inutile si la variable reçoit une valeur par saisie avant sa première utilisation.

ÉCRITURE

Syntaxe de l'instruction de lecture

```
saisir variable1 [, variable2, ...]
```

Exemples

- `saisir x` lit une valeur au clavier et l'affecte à x.
- `saisir x, y` lit deux valeurs au clavier. La première est affectée à x et la seconde à y.

LECTURE

Généralités sur la restitution des résultats

- La plupart des programmes doivent communiquer leurs résultats à l'utilisateur. Ils utilisent pour cela un périphérique de sortie, comme l'écran.
- L'utilisateur peut ainsi lire la valeur d'une variable, le résultat d'un calcul ou un message produit par le programme.

LECTURE

Syntaxe

```
afficher expression1 [, expression2, ...]
```

Exemples

- `afficher "Bonjour"` affiche le message `Bonjour`.
- `afficher x` affiche la valeur de x.
- `afficher "Somme = ", x + y` affiche `Somme = 15` si $x = 5$ et $y = 10$.



INSTITUT SABDARIFA

M1 Géomatique & Technologies numériques

Algorithmique et programmation

Chapitre 1 : Variables et instructions

M. Amadou MBAYE

E-mail : mbayeamadou@mbayeamadou.com

Site web : www.mbayeamadou.com

23 novembre 2026