

Chapitre 2 : Algorithmique

Durée du chapitre selon le programme scolaire : 2 semaines (8 heures de cours)

Objectif général : L'apprenant doit être capable de mobiliser les instructions élémentaires, telles que l'affectation et les opérations d'entrée-sortie, tout en choisissant les types de variables appropriés pour traduire un raisonnement mathématique en un script structuré. Il doit maîtriser l'usage des structures de contrôle, incluant les instructions conditionnelles et les boucles bornées ou non, ainsi que la définition de fonctions simples pour décomposer un problème complexe en sous-problèmes

Nom du créateur : Equipe maths

Date de création : Avril 2026

Sources de documents :

- <http://ressource.elec.free.fr/docs/NSI/algo/algo-partie1-variables%20et%20affectations.pdf>
- https://moodle.luniversitenumérique.fr/pluginfile.php/56967/mod_folder/content/0/Chapitre_6.pdf
- <https://www.mcours.net/cours/pdf/hascllic4/hassssclic373.pdf>
- https://www.lyceedadultes.fr/sitepedagogique/documents/math/math2S/00_algorithme/00_cours_algorithme_2025.pdf
- <https://coursessentiels.fr/documents/Algorithme%20et%20programmation%20nde%20Introduction%202018.pdf>

I. Introduction, histoire	
II. L'algorithme	

1. Définition	
2. Les quatres briques de base.....	
3. Les variables	
4. Les Operateurs.....	
5. Les Conditions (Si.....Alors...Sinon).....	
6. La boucle "Pour"	
7. Le boucle "Tant que"	
8. Les fonctions.....	
9. Le recherche dichotomique	
10. Algorithme d'euclide(PGCD)	

Mathématiques Seconde

I. INTRODUCTION, HISTOIRE ET APPROCHE PHYSIQUE

1. Origine historique de la notion

Les algorithmes existent pourtant bien avant lui :

- Les Babyloniens (vers 1800 av. J.-C.) utilisaient des méthodes algorithmiques pour résoudre des équations.
- Euclide (vers 300 av. J.-C.) a écrit un algorithme pour calculer le plus grand commun diviseur (PGCD), encore utilisé aujourd'hui.
- Al-Khwarizmi a systématisé et popularisé ces méthodes, ce qui a fait que son nom est resté dans l'histoire.

Au Moyen Âge, le mot latin "algorithmus" désignait l'art du calcul avec les chiffres arabes. Ce n'est qu'au XXe siècle, avec l'invention des ordinateurs, que le mot a pris son sens moderne : une série d'instructions

pour qu'une machine résolve un problème.

2. L'approche par physique

Avant d'être une notion informatique, l'algorithme est une démarche naturelle que l'on trouve en physique :

En mécanique : pour calculer la position d'un objet en mouvement, on applique étape par étape la formule $v = d/t$.

En électricité : pour savoir si une lampe s'allume, on suit un chemin logique (interrupteur fermé ? → circuit fermé ?).

En optique : pour tracer la marche d'un rayon lumineux, on applique des règles simples (réflexion, réfraction).

Un algorithme, c'est comme une recette de physique expérimentale : on suit un protocole précis pour obtenir un résultat mesurable.

II. L'algorithme

1. Définition

Un algorithme est une suite d'instructions simples et précises qui, exécutées dans l'ordre, permettent de résoudre un problème ou d'obtenir un résultat.

Il fonctionne comme une boîte noire :

- Entrée : des données fournies au départ
- Traitement : les instructions effectuées
- Sortie : le résultat final

Le mot vient du mathématicien persan Al-Khwarizmi (IXe siècle).

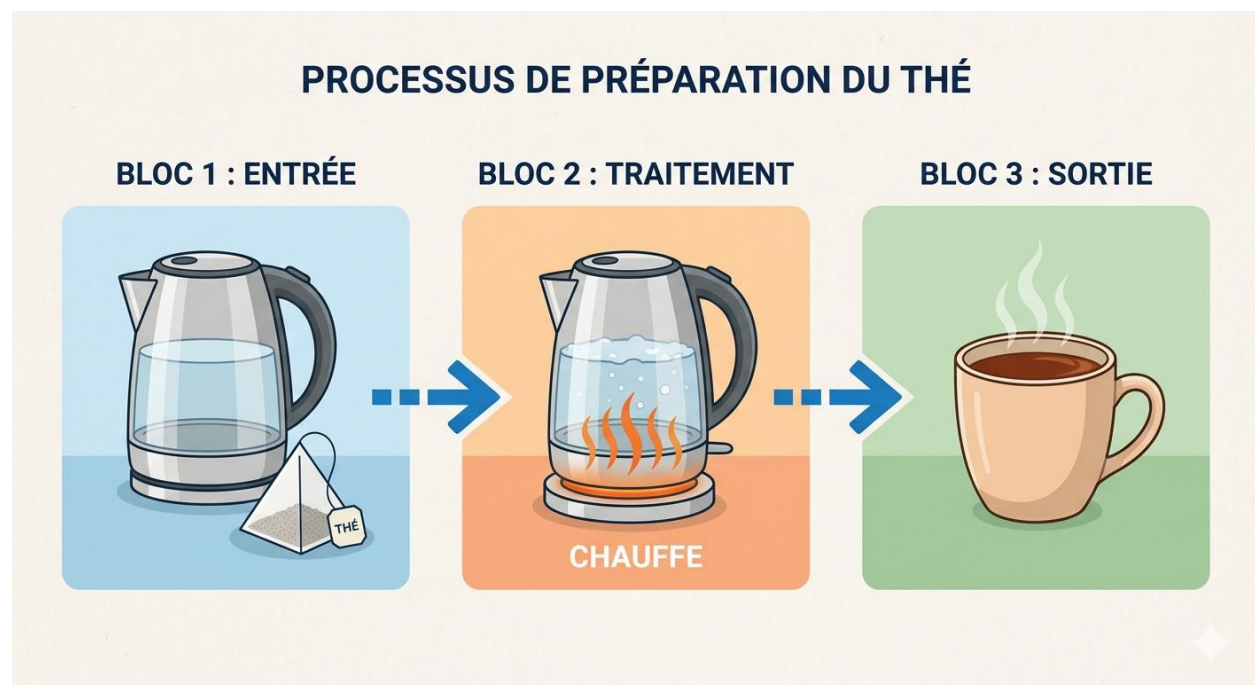
Exemple : préparer une tasse de thé

1. Mettre de l'eau dans la bouilloire
2. Allumez la bouilloire
3. Mettre un sachet de thé dans une tasse
4. Verser l'eau chaude dans la tasse
5. Laisser infuser 3 minutes
6. Retirer le sachet

Entrée : eau, sachet de thé

Traitement : les 6 étapes

Sortie : une tasse de thé



2. Les quatres briques de base

Tout algorithme repose sur quatre familles d'instructions :

1. **L'affectation** : stocker une valeur dans une variable
2. **L'entrée / la sortie** : lire des données ou afficher un résultat
3. **Le test conditionnel** : choisir entre plusieurs actions selon une condition
4. **La boucle** : répéter plusieurs fois une action

Exemple : algorithme pour dire si un nombre est pair

Afficher "Donne un nombre"

Lire nombre

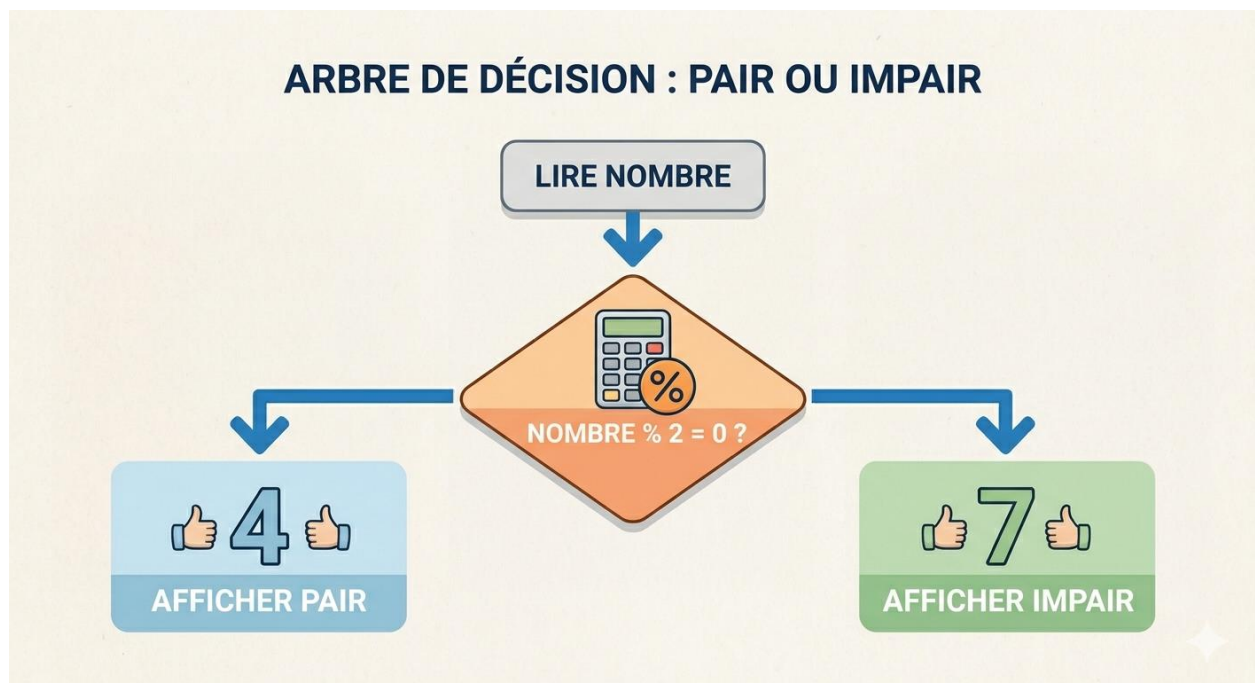
Si nombre $\% 2 = 0$ Alors

Afficher "Pair"

Sinon

Afficher "Impair"

Fin Si



3. Les variables

Une variable est un emplacement mémoire (une boîte étiquetée) contenant une valeur modifiable.

Types :

- Entier : 15, 42
- Décimal : 1.75, 3.14
- Chaîne : "Alice", "Bonjour"
- Booléen : Vrai, Faux

Affectation : on utilise la flèche <-

Exemple : échange de deux variables

A <- 3

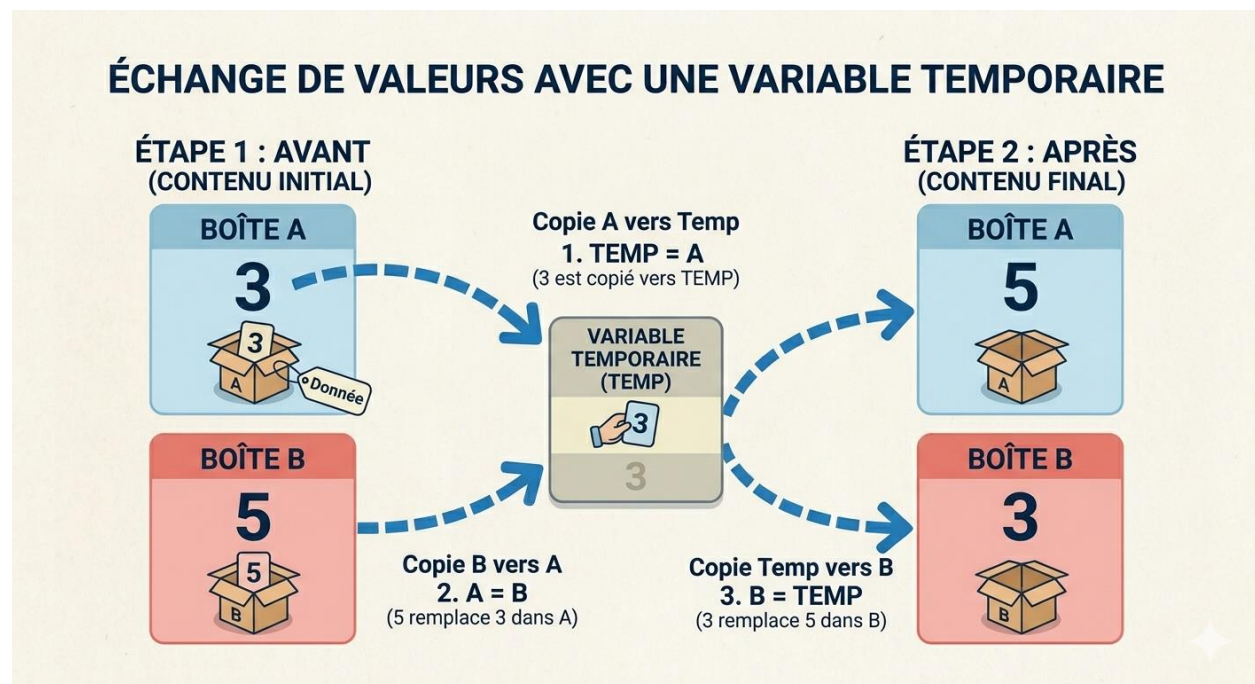
B <- 5

Temp <- A

A <- B

B <- Temp

Résultat final : A = 5, B = 3.



4. Les Operateurs

Les opérateurs sont des symboles qui permettent d'effectuer des calculs ou de faire des comparaisons entre des valeurs. Une expression est une combinaison de valeurs et d'opérateurs

(ex : $3 + 5$, $\text{age} \geq 18$, $(x > 0)$ ET $(y > 0)$). Ils servent à effectuer des calculs mathématiques sur des nombres.

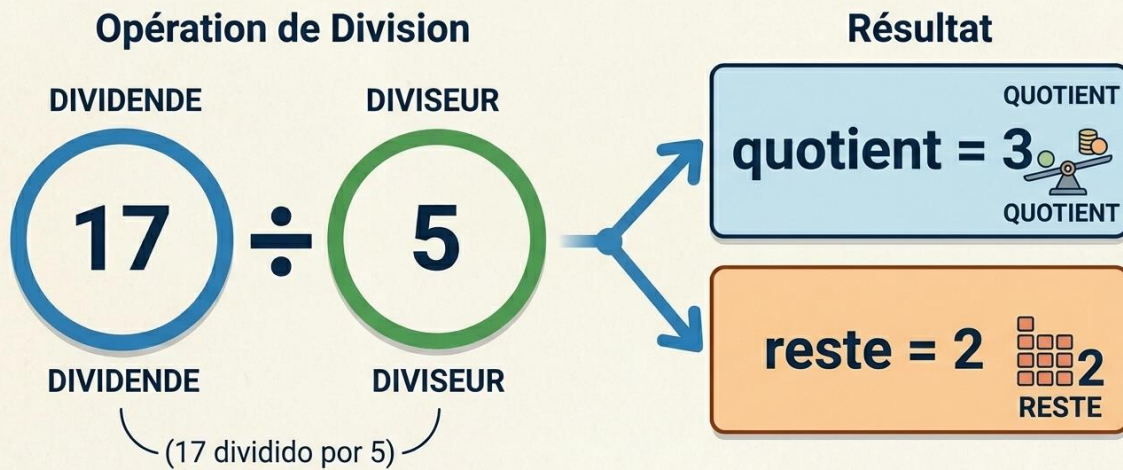
Operateur	Signification	Exemple	Resultat
+	addition	$15+2$	17
-	Soustraction	$15-2$	13
/	division	$15/2$	7.5
*	multiplication	$7*2$	14
//	Quotient entier	$15/2$	7
%	module	$7\%2$	1

Exemple : division et reste

$$17 \div 5 = 3 \text{ (quotient)}$$

$$\text{Reste} = 17 \% 5 = 2$$

DÉMONSTRATION DE LA DIVISION AVEC RESTE



5. Les Conditions (Si....Alors...Sinon)

Permet d'exécuter un bloc d'instructions seulement si une condition est vraie.

Exemple : majorité

Si age $\geq$ 18 Alors

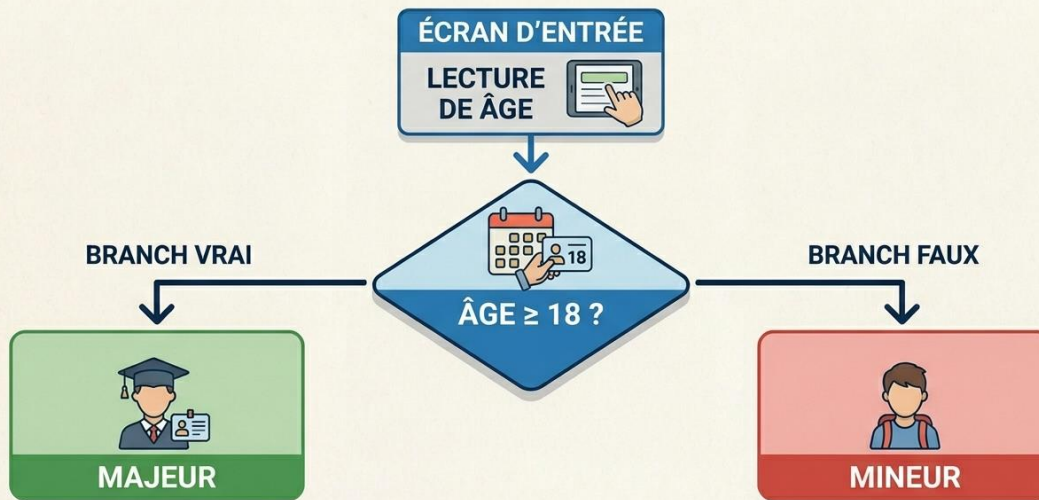
Afficher "Majeur"

Sinon

Afficher "Mineur"

Fin Si

ORGANIGRAMME DE DÉCISION : ÂGE MAJORITÉ



6. La boucle "Pour"

On connaît le nombre de répétitions à l'avance.

Exemple: somme des entiers de 1 à n

$S \leftarrow 0$

Pour i allant de 1 à n

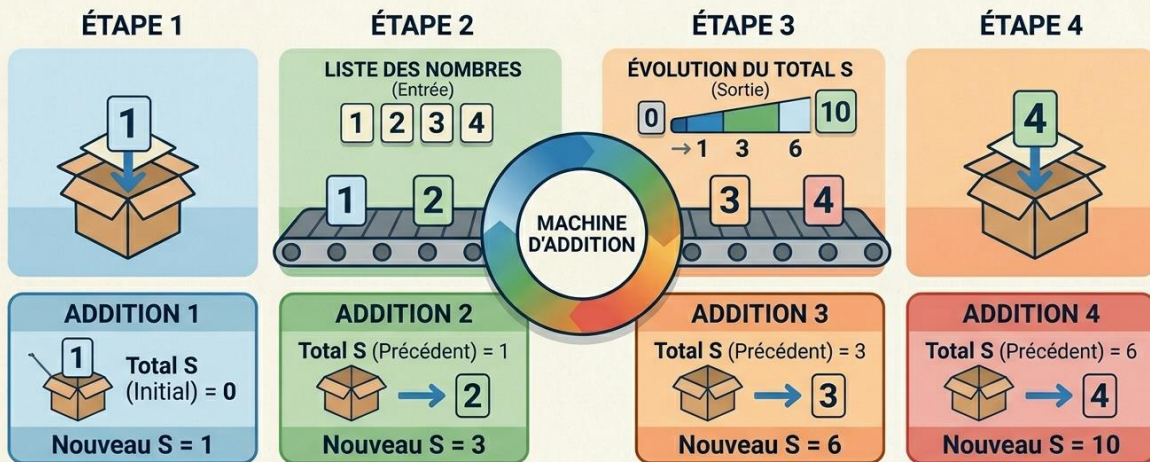
$S \leftarrow S + i$

Fin Pour

Afficher S

Avec $n = 4$: $S = 1+2+3+4 = 10$

FONCTIONNEMENT D'UN COMPTEUR D'ADDITION



CONCLUSION : À CHAQUE ÉTAPE, UN NOMBRE ENTRANT S'AJOUTE AU TOTAL S (PREMIER S=1, DEUXIÈME S=3, TROISIÈME S=6, QUATRIÈME S=10). LE TOTAL FINAL EST 10.

7. Le boucle "Tant que"

Répète tant qu'une condition est vraie. Attention aux boucles infinies.

Exemple : mot de passe

mdp ← ""

Tant que mdp ≠ "secret"

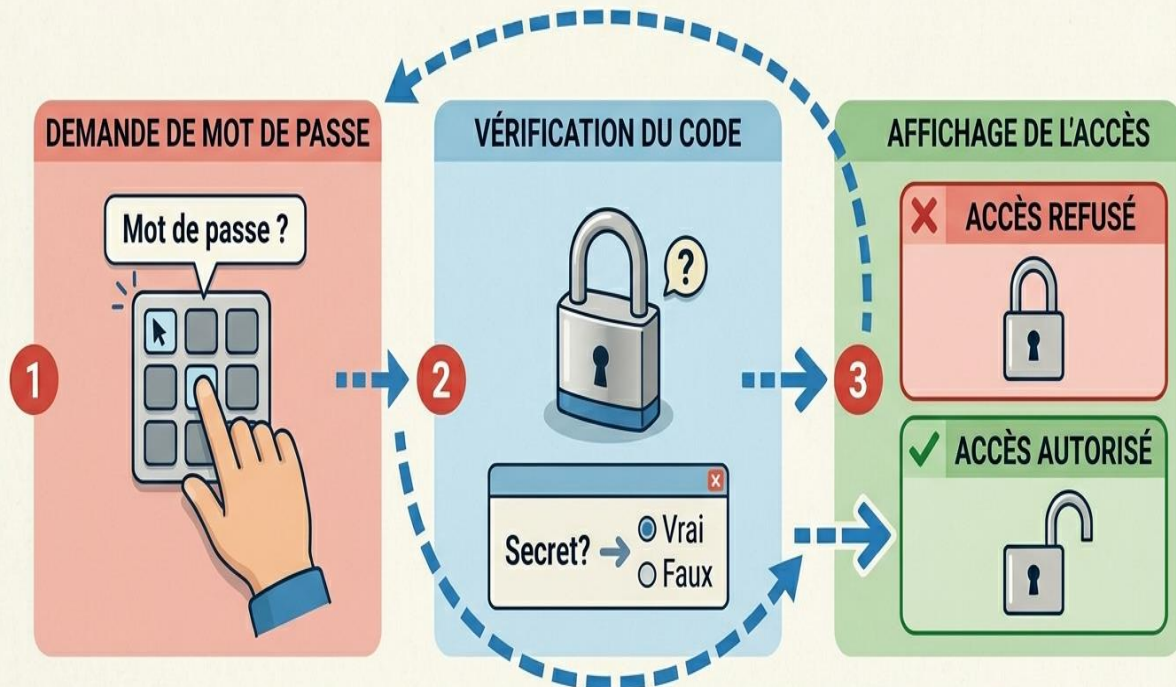
Afficher "Entrez le mot de passe"

Lire mdp

Fin Tant que

Afficher "Accès autorisé"

FONCTIONNEMENT D'UN SYSTÈME DE VÉRIFICATION DE MOT DE PASSE



REVIENT TOUJOURS À LA QUESTION TANT QUE "SECRET" N'EST PAS ENTRÉ
REVIENT TOUJOURS À LA QUESTION TANT QUE "SECRET" N'EST PAS ENTRÉ

8. Les fonctions

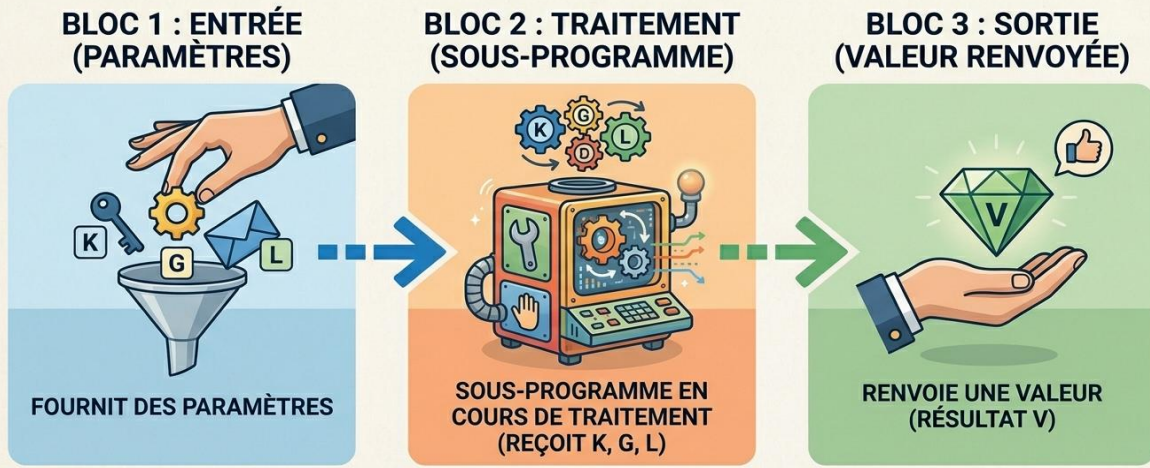
Définition

Une fonction est un **sous-programme** : un petit morceau d'algorithme indépendant, auquel on donne un nom, et que l'on peut **réutiliser** plusieurs fois.

Une fonction peut recevoir des données en entrée (appelées **paramètres** ou **arguments**) et **renvoie une valeur** (le résultat de son calcul).

On dit qu'une fonction "retourne" une valeur.

PROCESSUS DE FONCTION : SOUS-PROGRAMME ET RETOUR DE VALEUR



Vocabulaire à connaître

Définir une fonction : écrire son nom, ses paramètres et son contenu.

Appeler une fonction : l'utiliser dans un algorithme.

Paramètre : variable qui reçoit la valeur passée à la fonction.

Retourner : instruction qui donne le résultat de la fonction.

Structure generale de la fonction

Fonction Nom De La Fonction(paramètre 1, paramètre 2, ...)

instructions

Retourner valeur

Fin Fonction

Exemple : fonction Carré

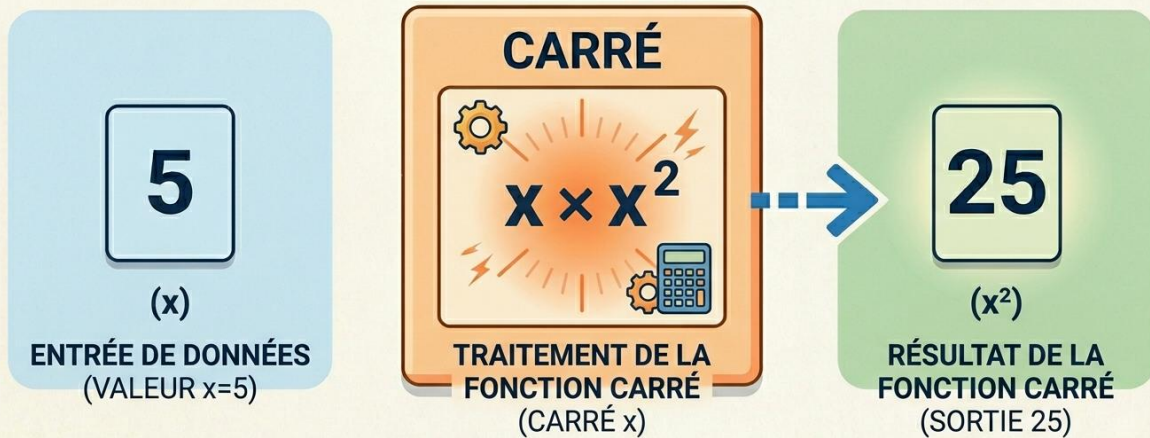
Fonction Carré(x)

Retourner $x \times x$

Fin Fonction

Utilisation : Carré(5) → 25

MODÉLISATION DE LA FONCTION CARRÉ ($x \rightarrow x^2$)



Pourquoi utiliser les fonctions

Les fonctions présentent plusieurs avantages :

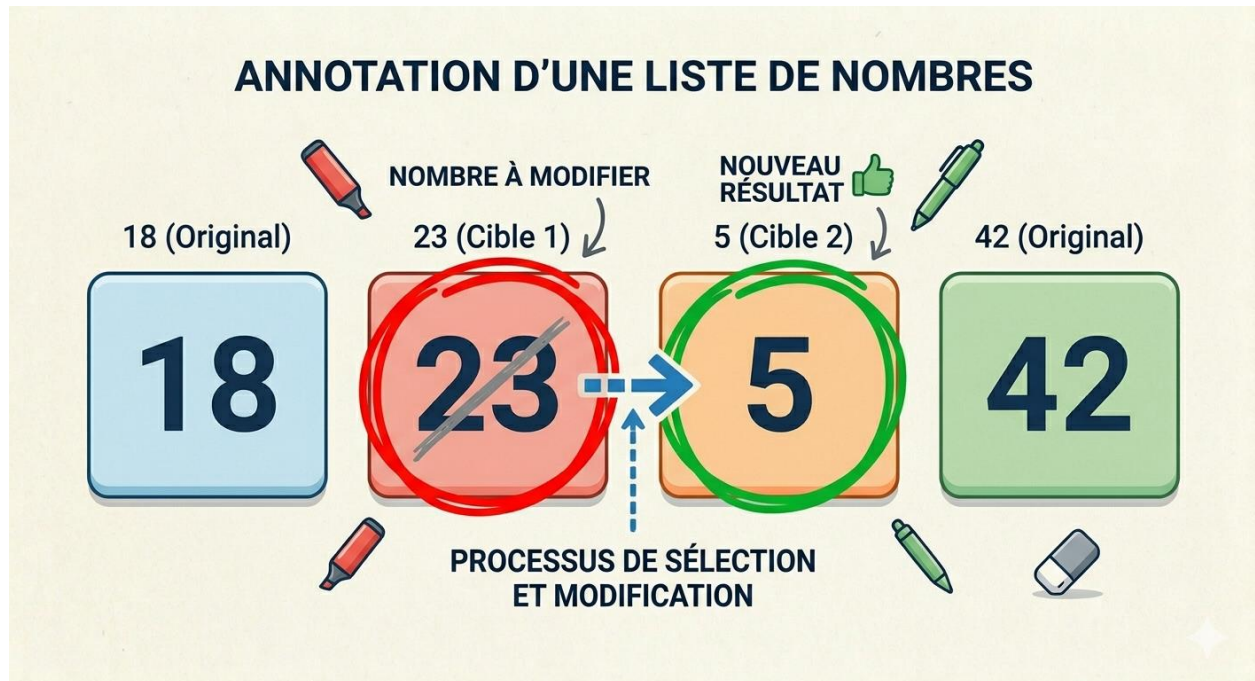
- Éviter la répétition: au lieu d'écrire 10 fois le même calcul, on écrit une seule fonction qu'on appelle 10 fois.
- Lisibilité: un programme avec des fonctions est plus clair et plus facile à comprendre.
- Réutilisation: une fois écrite, une fonction peut servir dans plusieurs algorithmes différents.
- Travail en équipe: différentes personnes peuvent écrire différentes fonctions, puis les assembler.

9. Le recherche dichotomique

Recherche rapide dans une liste triée. On compare avec le milieu et on élimine la moitié à chaque étape.

Exemple : chercher 5 dans [2, 5, 12, 23, 31, 45, 67]

- Étape 1 : milieu = 23. $5 < 23 \rightarrow$ on cherche dans la partie gauche [2, 5, 12]
- Étape 2 : nouveau milieu = 5. $5 = 5 \rightarrow$ trouvé en 2 étapes.



10. Algorithme d'euclide(PGCD)

Calcule le plus grand commun diviseur par divisions successives.

Exemple : PGCD(48, 18)

Tant que $B \neq 0$

$R \leftarrow A \% B$

$A \leftarrow B$

$B \leftarrow R$

Fin Tant que

Afficher A

DIAGRAMME DE CALCUL AVEC ÉCRASEMENT DE VALEURS (TABLEAU A, B, R)

