ALGORITHMIQUE · FICHE PRATIQUE

Cheatsheet : Données, opérateurs et bases de numération

Variables, opérateurs, priorités, logique, bases 2/8/10/16, nombres en machine, conversions et collections.

Par Ali ZAINOUL · Cours gratuit

Pour lire un programme, distinguez d'abord les données, les noms qui les désignent et les opérations effectuées. Cette fiche définit ces notions, puis montre comment suivre un changement d'état ligne par ligne.

L'état est l'ensemble des valeurs utiles à un instant du calcul. Les tableaux de trace ci-dessous donnent l'état après chaque instruction. Vous pouvez les refaire à la main avant d'exécuter les blocs.

1. Valeur, type, variable et constante

Une valeur peut être un entier, un nombre décimal, un texte ou une réponse logique. Son type indique sa nature et les opérations qui lui conviennent. Le nombre 4 et le texte "4" se ressemblent à l'écran, mais seul le premier est directement une donnée numérique.

Une variable est un nom associé à une valeur ; une affectation peut modifier cette association. Une constante nommée est une valeur que l'on s'engage à ne pas réaffecter pendant le traitement. SECONDS_PER_MINUTE = 60 exprime une règle stable ; total_seconds reçoit les durées successivement étudiées.

Des valeurs différentes

Notion	Exemple	Lecture
Entier	4	Nombre sans partie fractionnaire

Notion	Exemple	Lecture
Texte	"4"	Un caractère représentant un chiffre
Booléen	True / False en Python ; true / false en MATLAB	Réponse vraie ou fausse
Constante nommée	SECONDS_PER_MINUTE = 60	Convention de lisibilité, pas verrouillage automatique

2. Affectation et comparaison n’ont pas le même rôle

L'affectation évalue la partie droite, puis associe le résultat au nom de gauche. Après x = 2 puis x = x + 3, x vaut 5 : l'ancienne valeur est lue avant le remplacement. Une comparaison, elle, produit une réponse logique sans remplacer la valeur comparée.

Les parenthèses imposent un regroupement : 2 + 3 × 4 vaut 14, tandis que (2 + 3) × 4 vaut 20. Quand une expression semble ambiguë, écrivez son regroupement avant de la traduire.

Symboles des exemples scalaires

Rôle	Pseudo-code	Python	MATLAB
Affecter	x <- 2	x = 2	x = 2;
Comparer l’égalité	x == 2	x == 2	x == 2
Tester la différence	x != 2	x != 2	x ~= 2
Inférieur ou égal	x <= 2	x <= 2	x <= 2
ET logique	A AND B	A and B	A && B
OU / NON	A OR B / NOT A	A or B / not A	A || B / ~A

3. Échanger deux valeurs sans en perdre une

Écrire left = right puis right = left échoue : après la première ligne, les deux noms désignent déjà 9. Une variable temporaire conserve la valeur 4 avant son remplacement. Le raisonnement est identique dans les trois écritures.

Trace de l'échange

Instruction terminée	temporary	left	right
État initial	Non définie	4	9
temporary <- left	4	4	9
left <- right	4	9	9
right <- temporary	4	9	4

L’échange avec une variable temporaire

Les deux valeurs initiales sont left = 4 et right = 9. Le résultat attendu est left = 9 et right = 4.

Pseudo-code

```
// example.pseudo
// swap_values
left <- 4
right <- 9
temporary <- left
left <- right
right <- temporary
PRINT left, right
```

Python

```python
# swap_values.py
left = 4
right = 9
temporary = left
left = right
right = temporary
print(left, right)
```

`print` observe les valeurs finales ; il ne réalise pas l’échange.

MATLAB

```
% swap_values.m
left = 4;
right = 9;
temporary = left;
left = right;
right = temporary;
disp([left, right]);
```

`disp` affiche ici [9 4]. Le point-virgule évite les affichages automatiques intermédiaires.

> **Résultat expliqué.** Les noms left et right portent maintenant 9 et 4. La variable temporary peut encore contenir 4, mais l’échange est terminé.

4. Construire une condition d’intervalle

Une comparaison produit une réponse vraie ou fausse, appelée booléen. AND exige deux réponses vraies. OR accepte au moins une réponse vraie, y compris les deux. NOT inverse la réponse. Pour rester entre 18 et 24 degrés Celsius, bornes incluses, il faut satisfaire les deux comparaisons.

Les opérateurs MATLAB présentés ici portent sur des conditions scalaires, c'est-à-dire une seule réponse à la fois. Les opérations sur des tableaux logiques complets demandent d'autres conventions.

Une température dans la plage

Le programme doit accepter 18 et 24, mais refuser 17 et 25.

Pseudo-code

```
// example.pseudo
// check_range
temperature <- 21
in_range <- (temperature >= 18) AND (temperature <= 24)
PRINT in_range
```

Python

```python
# check_range.py
temperature = 21
in_range = (temperature >= 18) and (temperature <= 24)
print(in_range)
```

`and` impose les deux bornes. `True` signifie que 21 appartient à la plage.

MATLAB

```
% check_range.m
temperature = 21;
in_range = (temperature >= 18) && (temperature <= 24);
disp(in_range);
```

`disp` affiche ici 1 pour la valeur logique vraie. Le résultat de la comparaison est logique, même si son affichage ressemble au nombre 1.

> **Résultat expliqué.** 21 satisfait les deux bornes : le résultat logique est vrai.

> **Erreur à éviter.** Remplacer AND par OR accepterait aussi 10 ou 30 : chacune de ces températures vérifie au moins une des deux comparaisons.

▼ Après left = 4, right = 9, left = right et right = left, quelles sont les valeurs ? Quelle correction faut-il apporter ?

Les deux valeurs valent 9. Il faut conserver 4 dans temporary avant la première réaffectation de left, puis affecter temporary à right.

5. Opérations arithmétiques

Opérande : Une valeur sur laquelle agit un opérateur ; un opérateur unaire en utilise une, un binaire deux.

Division euclidienne : Pour $n \geq 0$ et $d > 0$, $n = q \times d + r$ avec $0 \leq r < d$.

Puissance : Python écrit ** ; MATLAB écrit ^ pour les scalaires.

Opérateurs arithmétiques

Opération	Python	MATLAB
Puissance	2 ** 3 = 8	2 ^ 3 = 8
Quotient / reste	17 // 5 = 3 ; 17 % 5 = 2	floor(17 / 5) = 3 ; mod(17, 5) = 2

Erreur à éviter. Python // arrondit vers moins l'infini, pas vers zéro.

▼ Quel quotient et quel reste pour 23 divisé par 4 ?

q = 5, r = 3 ; $23 = 5 \times 4 + 3$ et $0 \leq 3 < 4$.

6. Priorités et associativité

Priorité : La multiplication se regroupe avant l'addition ; les parenthèses imposent le regroupement.

Associativité : À priorité égale, elle détermine le regroupement des opérateurs.

Traduction : Les puissances se regroupent à droite en Python, à gauche en MATLAB.

Priorités et regroupements

Expression	Regroupement	Résultat
20 / 5 * 2	(20 / 5) * 2	8
2 ** 3 ** 2 (Python)	2 ** (3 ** 2)	512
2 ^ 3 ^ 2 (MATLAB)	(2 ^ 3) ^ 2	64

Erreur à éviter. Ne traduisez pas une chaîne de puissances sans expliciter ses parenthèses.

▼ **Quelle expression donne la moyenne de 8 et 12 ?**

(8 + 12) / 2 = 10 ; sans parenthèses, 8 + 12 / 2 = 14.

7. Bases 2, 8, 10 et 16

Base : En base b, les chiffres vont de 0 à b - 1 et les positions pèsent 1, b, b^2, etc.

En hexadécimal, A, B, C, D, E et F valent respectivement 10, 11, 12, 13, 14 et 15. Pour lire 1A, calculez 1 × 16 + 10 = 26. Pour lire 11010 en binaire, additionnez les poids 16, 8 et 2.

Conversion : Divisez par la base et lisez les restes en sens inverse ; zéro s'écrit 0.

Regroupement : Un chiffre octal correspond à trois bits ; un chiffre hexadécimal à quatre bits.

Écritures de vingt-six

Base	Écriture de vingt-six	Décomposition
2	11010	16 + 8 + 2
8	32	3 × 8 + 2
10	26	2 × 10 + 6
16	1A	1 × 16 + 10

Erreur à éviter. Une écriture comme 10 exige d'annoncer sa base ; la valeur ne change pas lors d'une conversion.

▼ Convertissez 101101_2 en décimal, octal et hexadécimal.

32 + 8 + 4 + 1 = 45 ; 101 101 donne 55_8 ; 0010 1101 donne $2D_{16}$.

8. Nombres en machine

Un bit est un chiffre binaire, 0 ou 1. Une suite de bits ne précise pas seule le type de la valeur représentée.

Octet : Huit bits offrent 256 configurations ; leur interprétation dépend du type.

Entier borné : Sur n bits : 0 à $2^n - 1$ non signé ; -2^{n-1} à $2^{n-1} - 1$ en complément à deux.

Flottant : Un nombre binaire de précision finie peut seulement approcher une fraction décimale. La somme 0.1 + 0.2 diffère ainsi légèrement de 0.3. Pour cet exemple, abs(result - 0.3) <= 1e-12 accepte cet écart ; une autre application doit choisir sa tolérance selon les unités et la précision requise.

Plages des entiers sur huit bits

Codage sur huit bits	Minimum	Maximum
Non signé	0	255
Complément à deux	-128	127

Erreur à éviter. Une tolérance dépend du problème et de ses unités ; l'affichage arrondi ne rend pas un calcul exact.

▼ Pourquoi 11111111 vaut-il parfois -1 ?

En complément à deux sur huit bits, le bit de gauche pèse -128 ; les autres valent ensemble 127. Sans signe, tous les poids sont positifs et donnent 255.

9. Conversion et validation

Conversion : Lire le texte "12" comme le nombre 12 précède son utilisation numérique.

Validation : Une conversion réussie ne garantit pas que la valeur appartienne au domaine autorisé.

Diagnostic : Distinguez syntaxe illisible, échec d'exécution et résultat logiquement incorrect.

Étapes du traitement d'une entrée

Étape	Exemple	Vérification
Lecture	"12"	Texte reçu
Conversion	12	Nombre obtenu
Validation	Entier non négatif	Règles du contrat
Affichage	3000 centimes	Unité annoncée

Erreur à éviter. Le texte "-2" se convertit, mais ne satisfait pas une quantité entière non négative.

▼ La conversion de "12.5" garantit-elle une quantité entière ?

Non. Python int échoue sur ce texte ; MATLAB str2double donne 12,5, qui n'est pas entier. Le type obtenu et le domaine métier doivent être vérifiés.

10. Texte et collections

Un élément est une valeur de la collection ; un indice désigne sa position. Pour [4, 7, 2], la deuxième valeur est 7 ; la remplacer par 9 donne [4, 9, 2] sans changer la longueur 3.

Collection : Une suite ordonnée possède des éléments ; sa longueur compte ces éléments.

Indice : Python et notre pseudo-code commencent à 0 ; MATLAB commence à 1.

Texte : Pour lire un caractère en MATLAB, le cours choisit le vecteur de caractères avec apostrophes.

Accès aux éléments

Opération	Python	MATLAB
Nombre d'éléments	len(values)	numel(values)
Premier élément	values[0]	values(1)
Deuxième élément	values[1]	values(2)

Erreur à éviter. Une collection vide n'a pas de premier élément : vérifiez sa longueur avant un accès.

▼ Quel indice désigne le dernier élément d'une collection de longueur 3 ?

2 en Python et dans notre pseudo-code ; 3 en MATLAB. Pour une longueur nulle, aucun indice ne permet de lire un élément.

Télécharger cette fiche en PDF · **Toutes les fiches par thème**

Version web : https://www.clearcode.fr/courses/programming/algorithmic/cheatsheet/foundations