ALGORITHMIQUE · CHAPITRE 9 · FONDATIONS

Organiser du texte et des suites de valeurs

Créez une chaîne et une collection, lisez leur longueur et leurs indices, puis modifiez un élément. Comparez les conventions Python et MATLAB.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Distinguer élément, indice, longueur et collection vide.
- Accéder à un élément et modifier une collection numérique.
- Lire un texte comme une suite de caractères simples.

Prérequis. Savoir affecter des valeurs et distinguer texte et nombre.

Comment retrouver puis corriger une valeur ?

Trois mesures, 4, 7 et 2, forment une suite ordonnée. Pour corriger seulement la deuxième, il faut pouvoir la désigner sans changer les autres. Une collection regroupe les valeurs ; un indice repère un emplacement.

Élément, indice et longueur

Un **élément** est une valeur de la collection. Un **indice** désigne une position ; ce n'est pas la valeur qui s'y trouve. La **longueur** est le nombre d'éléments. Notre suite contient trois éléments ; l'élément de deuxième position vaut 7.

Même suite, conventions différentes

Position	Indice Python / pseudo-code	Indice MATLAB	Valeur initiale

Position	Indice Python / pseudo-code	Indice MATLAB	Valeur initiale
Première	0	1	4
Deuxième	1	2	7
Troisième	2	3	2

Pour une longueur n, Python utilise ici les indices de 0 à n - 1, MATLAB de 1 à n ; notre pseudo-code suit Python. Une collection vide a une longueur nulle : elle n'a pas de premier élément. Tenter de le lire provoque une erreur dans ces langages.

Créer une collection et du texte

Python crée une **liste** avec `[4, 7, 2]`, lit sa longueur avec `len` et accède à un élément avec des crochets. MATLAB crée ici un **vecteur ligne numérique** avec `[4, 7, 2]`, compte ses éléments avec `numel` et utilise des parenthèses pour accéder à un élément. Notre pseudo-code choisit des crochets et `length`.

Une liste Python peut regrouper des valeurs de types différents ; le vecteur numérique MATLAB montré ici est homogène. Nous gardons des nombres dans les deux cas. Les écrire dans un même ordre ne signifie pas que tous les types de collections possèdent les mêmes règles.

Une chaîne Python `"code"` permet de lire le premier caractère avec `word[0]`. En MATLAB, nous choisissons explicitement le **vecteur de caractères** `'code'` : `word(1)` lit son premier caractère. Le texte MATLAB `"code"` serait au contraire un scalaire de type string ; `word(1)` désignerait alors ce string entier. Ces formes ne sont pas interchangeables.

Lire puis remplacer un élément

Pseudo-code

```
// example.pseudo
// access_values
// Positions start at 0 in this pseudocode.
values <- [4, 7, 2]
count <- length(values)
first <- values[0]
values[1] <- 9
word <- "code"
PRINT count, first, values[1]
```

Python

```python
# access_values.py
values = [4, 7, 2]
count = len(values)
first = values[0]
values[1] = 9
word = "code"
letter = word[0]
print(count, first, values[1], letter)
```

MATLAB

```
% access_values.m
values = [4, 7, 2];
count = numel(values);
first = values(1);
values(2) = 9;
word = 'code';
letter = word(1);
disp([count, first, values(2)]);
disp(letter);
```

La longueur reste 3, la première valeur reste 4, et la deuxième devient 9 ; le caractère lu est c. Une affectation à un élément modifie cet élément, pas tous les éléments de la collection. Les chaînes Python sont immuables : on ne peut pas remplacer un caractère par une affectation à un indice comme pour une liste.

Vérifier les frontières d’accès

Prédisez les indices du dernier élément : 2 en Python, 3 en MATLAB. Remplacez ensuite la collection par `[]`. Sa longueur vaut 0 ; la lecture du premier élément échoue. Dans le chapitre sur les conditions, nous apprendrons à tester ce cas avant un accès. Pour le texte, restez ici aux caractères simples du mot code ; les caractères Unicode composés demandent des précautions supplémentaires.

Préparer les parcours

▼ Après la correction, values[1] désigne-t-il 4 ou 9 ?

En Python, 9 : l’indice 1 est la deuxième position. Dans notre pseudo-code, 9 également : il commence à 0 comme Python. Il faut annoncer la convention avant de traduire.

▼ Faut-il connaître les boucles pour lire une collection ?

Non. On peut déjà créer la collection et lire un élément précis. Une boucle permettra ensuite de répéter une action pour chacun de ses éléments, en respectant les bornes et le cas vide.

À retenir - fiche mémo

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.

Pseudo-code

```
// example.pseudo
// access_values
values <- [4, 7, 2]
first <- values[0]
values[1] <- 9
```

Point de vigilance. Une collection vide n’a pas de premier élément : vérifiez sa longueur avant un accès.

Consulter la cheatsheet : Fondations · **Télécharger la fiche du thème en PDF**

Version web : https://www.clearcode.fr/courses/programming/algorithmic/foundations/sequences-and-strings