ALGORITHMIQUE · CHAPITRE 13 · CONDITIONS ET BOUCLES

Combiner des décisions

Lisez des conditions imbriquées, déterminez quels tests sont exécutés et simplifiez une règle de tarif sans changer son résultat.

Par Ali ZAINOUL · Cours gratuit

Ce que vous allez apprendre

- Identifier la condition qui autorise l'entrée dans un sous-bloc.
- Tracer une décision à deux niveaux.
- Comparer une imbrication et une chaîne de conditions.

Prérequis. Savoir lire des conditions simples et plusieurs branches.

Une règle de tarif fictive

Pour cet exercice, le tarif est de 5 unités avant 12 ans. À partir de 12 ans, il est de 7 unités pour les étudiants et de 10 pour les autres. L'âge est un entier non négatif et `student` est un booléen. Le statut étudiant n'intervient donc pas pour une personne de 11 ans.

Définir une condition imbriquée et ses entrées

Une **condition imbriquée** est un choix situé à l'intérieur d'une branche d'un autre choix. Pour l'atteindre, l'exécution doit d'abord entrer dans cette branche. Le programme reçoit ici `age`, un âge entier non négatif, et `student`, un booléen indiquant le statut étudiant. Le mot anglais `student` est un nom de variable ; il ne réalise pas lui-même une vérification administrative.

La règle fictive choisie pour l'exercice est entièrement précisée : avant 12 ans, le tarif vaut 5 unités monétaires ; à partir de 12 ans, il vaut 7 pour un étudiant et 10 sinon. Ces montants servent uniquement d'exemple. Pour un enfant de 8 ans, la réponse ne dépend pas du statut étudiant : le test intérieur n'est pas exécuté.

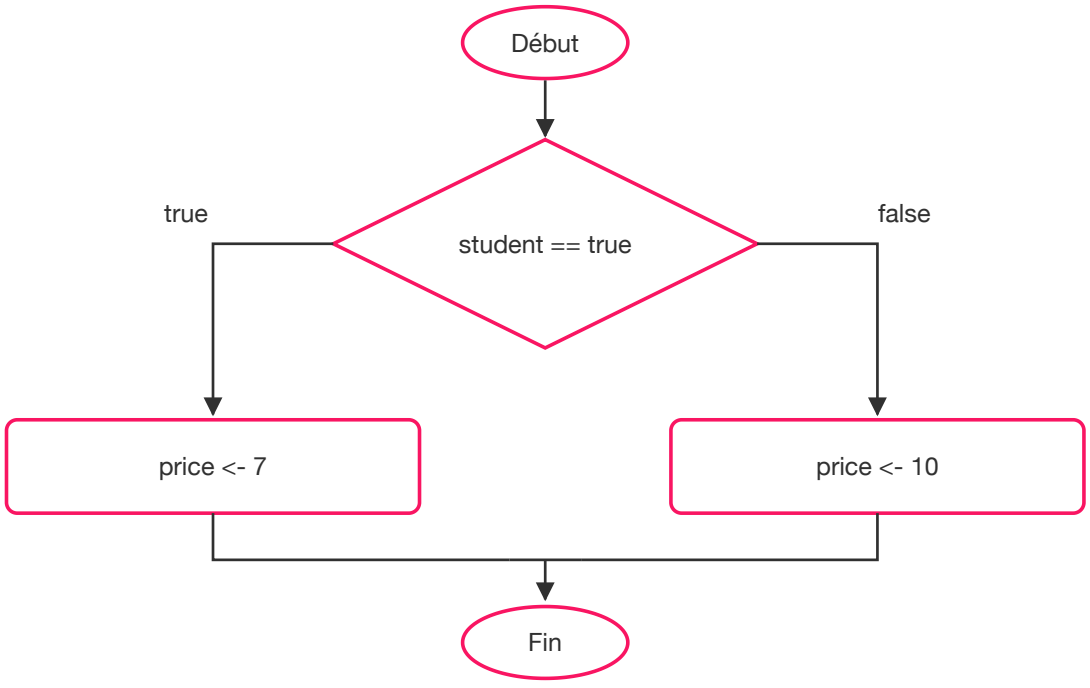
Un **chemin d'exécution** est la succession des instructions réellement parcourues pour une entrée. Avec 16 ans et `student = true`, on évite la première branche, on entre dans la seconde, puis on choisit 7. Avec 16 ans et faux, le même premier choix conduit au test intérieur, mais celui-ci choisit 10.

Le diagramme de ce chapitre représente seulement la seconde partie : il suppose déjà que l'âge est au moins 12 ans. Un schéma partiel doit annoncer cette hypothèse. Sinon, le lecteur pourrait appliquer le tarif étudiant à un enfant et croire que la représentation contredit le code complet.

Lire un bloc dans son contexte

On choisit d'abord le groupe d'âge. Dans la branche "12 ans ou plus", on effectue ensuite le test du statut étudiant. La condition extérieure constitue le contexte du test intérieur : ce dernier n'est pas exécuté pour toutes les entrées.

Ce sous-diagramme s'applique seulement après avoir établi que l'âge est au moins 12 ans.



▼ **Lire les parcours sans animation**

age = 16, student = true

1. Début du parcours avec age = 16, student = true. Les données sont déjà disponibles.
2. Test 1 : student == true donne true (vrai).
3. Action choisie : price <- 7.
4. Les chemins alternatifs se rejoignent. Une seule branche a été exécutée ; il n’y a pas deux tâches à attendre.
5. Fin du parcours. Résultat observable : price = 7.

age = 16, student = false

1. Début du parcours avec age = 16, student = false. Les données sont déjà disponibles.
2. Test 1 : student == true donne false (faux).
3. Action choisie : price <- 10.
4. Les chemins alternatifs se rejoignent. Une seule branche a été exécutée ; il n’y a pas deux tâches à attendre.
5. Fin du parcours. Résultat observable : price = 10.

Deux niveaux d’indentation

Pseudo-code

```
// example.pseudo
// choose_price
age <- 16
student <- TRUE
IF age < 12 THEN
    price <- 5
ELSE
    IF student THEN
        price <- 7
    ELSE
        price <- 10
    END IF
END IF
PRINT price
```

Python

```python
# choose_price.py
age = 16
student = True
if age < 12:
    price = 5
else:
    if student:
        price = 7
    else:
        price = 10
print(price)  # 7
```

MATLAB

```
% choose_price.m
age = 16;
student = true;
if age < 12
    price = 5;
else
    if student
        price = 7;
    else
        price = 10;
    end
end
disp(price); % 7
```

Chaque SINON doit être associé au bon SI. L'indentation Python matérialise ces niveaux ; les deux `end` MATLAB ferment successivement le test intérieur et le test extérieur. Le commentaire sur les tarifs fictifs rappelle que nous étudions une règle choisie pour l'exercice.

Construire les chemins

Quatre entrées pour distinguer les deux niveaux

age	student	Test intérieur ?	price
11	faux	Non	5
11	vrai	Non	5
12	vrai	Oui	7
12	faux	Oui	10

Suivez une ligne du tableau sans sauter directement au tarif attendu. Dites d'abord quelle branche extérieure est prise, puis si un second test est nécessaire. Cette verbalisation révèle souvent une mauvaise association entre SI et SINON.

Réduire l'imbrication

▼ Peut-on exprimer cette règle avec SI / SINON SI / SINON ?

Oui : SI age < 12, tarif 5 ; SINON SI student, tarif 7 ; SINON, tarif 10. Le second test est toujours atteint seulement pour un âge d'au moins 12 ans. Les deux écritures peuvent donc conserver le même comportement.

▼ Pourquoi ne pas commencer par tester student ?

Un étudiant de 11 ans pourrait recevoir le tarif 7 au lieu de 5 si l'on n'exprime pas correctement la priorité du critère d'âge. La simplification doit préserver la règle, pas seulement raccourcir le texte.

Garder peu de niveaux à la fois

Une imbrication n'est pas une erreur : elle peut rendre visible un contexte. Lorsqu'elle devient difficile à suivre, cherchez une condition plus claire ou une sous-tâche à isoler. Le cours sur les fonctions fournira un autre outil pour cette décomposition.

À retenir - fiche mémo

Imbrication

Un choix intérieur est exécuté seulement si la branche qui le contient est atteinte.

Chemin

Pour moins de 12 ans, le tarif est 5 sans consulter le statut étudiant.

Contrat

À partir de 12 ans, le statut étudiant distingue les tarifs 7 et 10.

Pseudo-code

```
// example.pseudo
// choose_price
IF age < 12 THEN
    price <- 5
ELSE IF student THEN
    price <- 7
ELSE
    price <- 10
END IF
```

Point de vigilance. Un schéma partiel doit annoncer l'hypothèse qui permet d'y entrer.

Consulter la cheatsheet : Conditions et boucles · **Télécharger la fiche du thème en PDF**

Version web : https://www.clearcode.fr/courses/programming/algorithmic/control-flow/conditions/nested-conditions