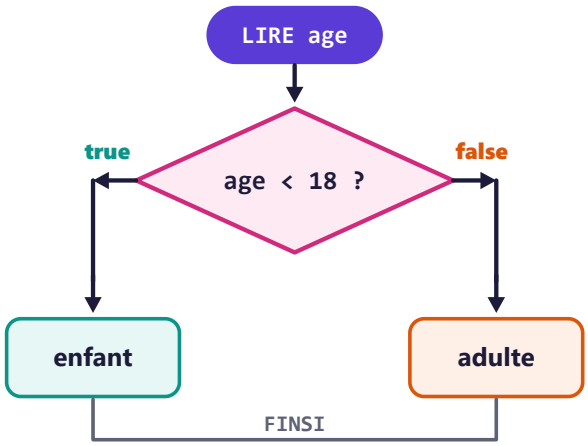
CHAPITRE 3

Les conditions : SI, SINON, SINONSI

Quand l'algorithme doit choisir



AU PROGRAMME

1. Prendre une décision : SI
2. Qu'est-ce qu'une condition ?
3. Plusieurs cas : SINONSI
4. Les conditions composées
5. Booléens et tests imbriqués

Nom : ..... **Classe :** .....

AVANT DE COMMENCER

Ce que tu vas apprendre

Jusqu'ici, ton algorithme exécutait toutes ses instructions, de haut en bas. Avec les tests, il devient capable de choisir : afficher « accès autorisé » ou « mot de passe incorrect », appliquer un tarif réduit ou le plein tarif. C'est la troisième famille d'instructions.

1. **Écrire un test avec `SI … ALORS … FINSI`, puis avec `SINON`.**
2. **Définir une condition et utiliser les six opérateurs de comparaison.**
3. **Enchaîner plusieurs cas avec `SINONSI`.**
4. **Combiner des conditions avec `ET`, `OU`, `XOR` et `NON`.**
5. **Utiliser une variable booléenne dans un test.**
6. **Imbriquer un test dans un autre quand c'est nécessaire.**

PARTIE 1

Prendre une décision : SI

Quand tu te connectes à Windows, l'ordinateur compare ce que tu as tapé avec ce qui est enregistré. **Si** c'est correct, il ouvre ta session ; **sinon**, il affiche une erreur. Ce choix entre deux chemins, c'est un **test**.

PSEUDOCODE La structure de base

```
SI condition ALORS
    instructions
FINSI
```

La condition vaut `true` ou `false`. Si elle est **vraie**, les instructions entre `ALORS` et `FINSI` sont exécutées. Si elle est **fausse**, l'algorithme les saute et continue après le `FINSI`.

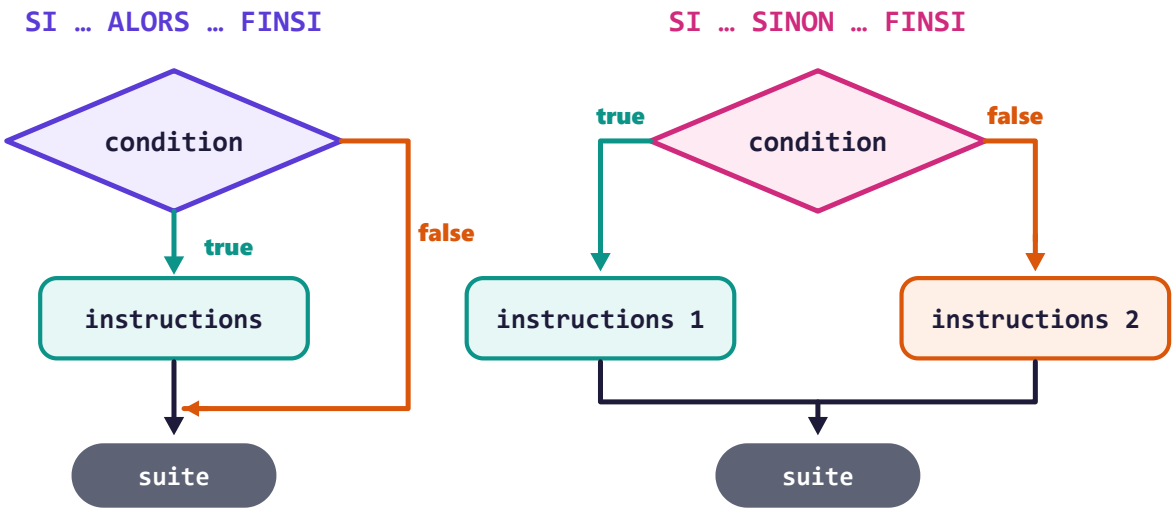


Figure 3.1 — À gauche, le SI seul ; à droite, le SI avec un SINON : deux chemins qui se rejoignent.

Deux chemins : SINON

PSEUDOCODE

```
SI condition ALORS
    instructions 1
SINON
    instructions 2
FINSI
```

Si la condition est vraie, l'algorithme exécute les instructions 1 ; si elle est fausse, il exécute les instructions 2. **Jamais les deux.** Dans tous les cas, il reprend ensuite après le `FINSI`.

À RETENIR

La mise en page

- Chaque `SI` a son `FINSI`.
- On décale les instructions à l'intérieur : on voit d'un coup d'œil ce qui dépend du test.

PARTIE 2

Qu'est-ce qu'une condition ?

DÉFINITION

Une condition

Une **comparaison**. Elle est formée de trois éléments : une valeur, un opérateur de comparaison et une autre valeur. À un moment donné, elle est soit vraie, soit fausse.



Figure 3.2 — Trois éléments, un résultat : true ou false.

Comparaison	En pseudocode	Exemple vrai
égal à	`=`	`3 = 3`
différent de	`<>`	`3 <> 5`
strictement plus petit	`<`	`3 < 5`
strictement plus grand	`>`	`5 > 3`
plus petit ou égal	`<=`	`3 <= 3`
plus grand ou égal	`>=`	`5 >= 3`

ATTENTION

Comparer ce qui est comparable

Les deux valeurs comparées doivent être **du même type**. Comparer 2 à 10 a un sens ; comparer 2 au mot "carotte" n'en a aucun et provoque une erreur.

PSEUDOCODE **Enfant ou adulte ?**

```
variable age en entier
DEBUT
    ECRIRE "Quel âge as-tu ?"
    LIRE age
    SI age < 18 ALORS
        ECRIRE "Catégorie enfant"
    SINON
        ECRIRE "Catégorie adulte"
    FINSI
FIN
```

À L'ÉCRAN

```
Quel âge as-tu ?
15
Catégorie enfant
```

Le `=` d'une condition **compare** ; il ne range rien. Ne le confonds pas avec la flèche `<-` de l'affectation.

PARTIE 3

Plusieurs cas : SINONSI

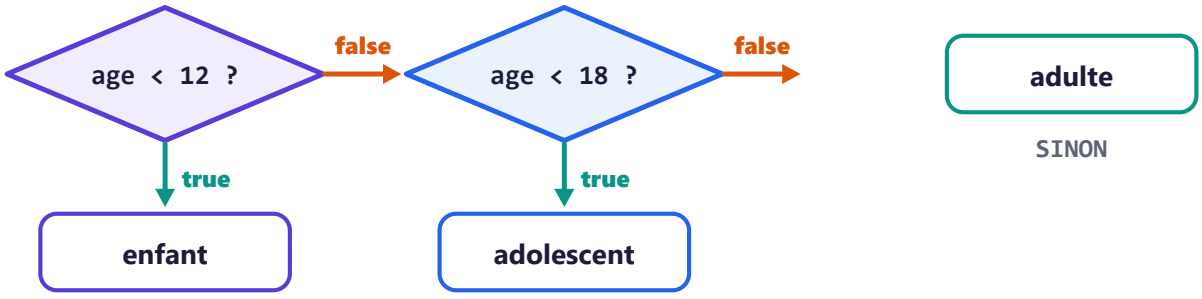
Un `SI … SINON` ne propose que deux chemins. Pour trois catégories ou plus, on ajoute des `SINONSI` : chacun teste une nouvelle condition, **seulement si toutes les précédentes étaient fausses**.

PSEUDOCODE

```
variable age en entier
DEBUT
    ECRIRE "Quel âge as-tu ?"
    LIRE age
    SI age < 12 ALORS
        ECRIRE "Catégorie enfant"
    SINONSI age < 18 ALORS
        ECRIRE "Catégorie adolescent"
    SINON
        ECRIRE "Catégorie adulte"
    FINSI
FIN
```

À L'ÉCRAN

```
Quel âge as-tu ?
14
Catégorie adolescent
```



Le premier test vrai l'emporte : les suivants ne sont même pas évalués.

Figure 3.3 — Les tests sont essayés dans l'ordre ; le SINON attrape tout le reste.

Au `SINONSI age < 18`, on sait déjà que l'âge n'est **pas** inférieur à 12 (sinon on serait entré dans le premier cas) : inutile d'écrire `age >= 12 ET age < 18`. L'**ordre** des tests

compte donc.

À RETENIR

La règle

- Des cas qui s'excluent (tranches d'âge, notes, tarifs) s'écrivent avec une chaîne `SI … SINONSI … SINON`, jamais avec des `SI` enfermés dans des `SINON`.
- Un seul `FINSI` ferme toute la chaîne.
- Le `SINON` final est facultatif : il traite « tous les autres cas ».

PARTIE 4

Les conditions composées

La phrase « mon âge est compris entre 20 et 30 » est tentante à traduire par `20 < age < 30`. C'est **faux** en programmation : deux opérateurs et trois valeurs, c'est une comparaison de trop.

Cette phrase contient en réalité **deux** conditions : l'âge est plus grand que 20 **et** l'âge est plus petit que 30. On les relie avec un **opérateur logique**.

PSEUDOCODE

```
SI age > 20 ET age < 30 ALORS
```

Opérateur	Le résultat est vrai…
`ET`	si les **deux** conditions sont vraies, et seulement dans ce cas.
`OU`	si **au moins une** condition est vraie (y compris les deux).
`XOR`	si **une seule** des deux est vraie : c'est le « ou exclusif ». Les deux vraies donnent faux.
`NON`	si la condition est fausse : il **inverse** le résultat. On l'écrit devant des parenthèses : `NON(age < 18)`.

La table de vérité

A	B	A ET B	A OU B	A XOR B
true	true	true	true	false
true	false	false	true	true
false	true	false	true	true
false	false	false	false	false

EXEMPLE

Exemple : le XOR

Au restaurant, le menu à 9 € comprend le plat avec **soit** une entrée, **soit** un dessert, mais pas les deux : `SI entree = "o" XOR dessert = "o" ALORS`.

POUR COMPRENDRE

Le OU de tous les jours est souvent exclusif : « fromage ou dessert » veut dire l'un **ou** l'autre. Le `OU` informatique, lui, accepte aussi les deux à la fois ; le ou exclusif, c'est `XOR`.

PARTIE 5

Booléens et tests imbriqués

Une variable booléenne dans un test

Une variable `booleen` vaut déjà `true` ou `false` : elle peut servir **directement** de condition, sans rien comparer. On peut aussi y ranger le résultat d'une condition.

PSEUDOCODE

```
variable age en entier
variables majeur, permis en booleen
DEBUT
    ECRIRE "Quel âge as-tu ?"
    LIRE age
    ECRIRE "As-tu ton permis ? (true ou false)"
    LIRE permis
    majeur <- age >= 18
    SI majeur ET permis ALORS
        ECRIRE "Tu peux conduire seul."
    SINON
        ECRIRE "Tu ne peux pas conduire seul."
    FINSI
FIN
```

À L'ÉCRAN

```
Quel âge as-tu ?
19
As-tu ton permis ? (true ou false)
True
Tu peux conduire seul.
```

`majeur <- age >= 18` range dans `majeur` le résultat de la comparaison : `true` ou `false`. Un nom de booléen bien choisi rend le test lisible comme une phrase.

Un test dans un test

À l'intérieur d'un `SI`, d'un `SINONSI` ou d'un `SINON`, on peut placer un autre test. On le fait quand la deuxième question **ne se pose que** dans un des cas.

PSEUDOCODE

```
variable note en entier
variable rattrapage en booleen
DEBUT
    ECRIRE "Ta note sur 20 ?"
    LIRE note
    SI note >= 10 ALORS
        ECRIRE "Réussi !"
    SINON
        ECRIRE "As-tu droit au rattrapage ? (true ou false)"
        LIRE rattrapage
        SI rattrapage ALORS
            ECRIRE "Rendez-vous en août."
        SINON
            ECRIRE "Il faudra recommencer l'année."
        FINSI
    FINSI
FIN
```

À L'ÉCRAN

```
Ta note sur 20 ?
8
As-tu droit au rattrapage ? (true ou false)
True
Rendez-vous en août.
```

ATTENTION

Un FINSI par SI

Chaque test imbriqué a **son propre** `FINSI`. Décale bien ton code : c'est le seul moyen de voir quel `FINSI` ferme quel `SI`.

BILAN DU CHAPITRE 3

L'essentiel à retenir

1. Un **test** permet de choisir : `SI condition ALORS … FINSI` exécute les instructions seulement si la condition est vraie.
2. Avec `SINON`, on exécute un bloc ou l'autre, jamais les deux, puis on reprend après le `FINSI`.
3. Une **condition** est une comparaison : une valeur, un opérateur (`=`, `<>`, `<`, `>`, `<=`, `>=`) et une valeur du même type.
4. Pour des cas qui s'excluent, on enchaîne `SI … SINONSI … SINON … FINSI` : le premier test vrai l'emporte.
5. `20 < age < 30` est faux : on écrit `age > 20 ET age < 30`.
6. `ET` : les deux vraies. `OU` : au moins une. `XOR` : une seule. `NON(…)` inverse.
7. Un **booléen** peut servir directement de condition ; un test peut être **imbriqué** dans un autre, avec son propre `FINSI`.

Le lexique

Test
Instruction qui choisit entre plusieurs chemins selon une condition (`SI`).

Condition
Comparaison qui vaut `true` ou `false` .

Opérateur de comparaison
`=` , `<>` , `<` , `>` , `<=` , `>=` .

Opérateur logique
`ET` , `OU` , `XOR` , `NON` : ils relient ou inversent des conditions.

Condition composée
Plusieurs conditions reliées par un opérateur logique.

XOR
Le « ou exclusif » : vrai si une seule des deux conditions est vraie.

SINONSI
Teste un nouveau cas quand tous les précédents étaient faux.

Test imbriqué
Un test placé à l'intérieur d'un autre test.

Je fais le point

Coche une case par ligne : où en es-tu avant l'interro ?

Je suis capable de…	Oui	Presque	Pas encore
Écrire un `SI … SINON … FINSI` correctement indenté.	☐	☐	☐
Choisir le bon opérateur de comparaison.	☐	☐	☐
Enchaîner plusieurs cas avec `SINONSI` dans le bon ordre.	☐	☐	☐
Traduire « entre 20 et 30 » en condition composée.	☐	☐	☐
Prévoir le résultat de `ET` , `OU` , `XOR` et `NON` .	☐	☐	☐
Utiliser un booléen comme condition.	☐	☐	☐

CHAPITRE 3 · EXERCICES

Teste tes connaissances

avant l'interro

COMMENT T'Y PRENDRE

1. **Relis la synthèse** : l'essentiel, le lexique et « Je fais le point ».
2. **Fais les exercices sans le cours**, comme à l'interro.
3. **Corrige-toi** avec la correction vue en classe ou sur le site.
4. **Reviens sur tes erreurs** : relis la partie du cours concernée.

LES 6 EXERCICES

1. **Les conditions**
2. **La table de vérité**
3. **Vrai ou faux pour ces valeurs ?**
4. **Que va-t-il s'afficher ?**
5. **Les bonnes pratiques**
6. **Mes premiers tests**

Exercice 1 Les conditions

▸ **Coche la bonne réponse.**

1. Que fait l'algorithme si la condition d'un `SI` sans `SINON` est fausse ?

- ☐ il s'arrête
- ☐ il saute au FINSI et continue
- ☐ il exécute quand même les instructions

2. Quel opérateur signifie « différent de » ?

- ☐ !=
- ☐ <>
- ☐ =/
- ☐ ><

3. `age >= 18` est vraie quand age vaut...

- ☐ 17
- ☐ 18
- ☐ aucun des deux

4. `true XOR true` vaut...

- ☐ true
- ☐ false

Exercice 2 La table de vérité

▸ **Complète le tableau avec true ou false.**

A	B	A ET B	A OU B	A XOR B	NON(A)
true	true				
true	false				
false	true				
false	false				

Exercice 3 Vrai ou faux pour ces valeurs ?

- La variable age vaut 16 et ville vaut "Waremme". Écris true ou false.

Condition	Résultat
`age > 12 ET age < 18`	
`age < 12 OU ville = "Huy"`	
`age >= 16 XOR ville = "Waremme"`	
`NON(ville = "Huy")`	
`age <> 16`	

Exercice 4 Que va-t-il s'afficher ?

- **Lis l'algorithme, puis réponds.**

PSEUDOCODE

```
variable note en entier
DEBUT
    ECRIRE "Note sur 20 ?"
    LIRE note
    SI note >= 14 ALORS
        ECRIRE "Bien"
    SINONSI note >= 10 ALORS
        ECRIRE "Suffisant"
    SINON
        ECRIRE "Insuffisant"
    FINSI
FIN
```

1. Qu'affiche l'algorithme si l'utilisateur tape 15 ?

.....

2. Qu'affiche-t-il si l'utilisateur tape 7 ?

.....

3. Qu'affiche-t-il si l'utilisateur tape 10 ?

.....

Exercice 5 Les bonnes pratiques

- **Coche vrai ou faux. Si c'est faux, corrige la phrase sur la ligne.**

Affirmation	Vrai	Faux
`SI 10 < temperature < 25 ALORS` est une condition correcte.	☐	☐
Une chaîne `SI … SINONSI … SINON` se ferme par un seul `FINSI`.	☐	☐
Dans `SI note = 20 ALORS`, le `=` range 20 dans note.	☐	☐
Si `connecte` est un booléen, on peut écrire `SI connecte ALORS`.	☐	☐
`A OU B` est faux quand A et B sont vraies toutes les deux.	☐	☐

Exercice 6 Mes premiers tests

- **Écris l'algorithme complet : déclarations, DEBUT, instructions, FIN.**

1. Demande un nombre entier et affiche s'il est pair ou impair.

EXEMPLE D'ÉCRAN

```
Entre un nombre entier :
7
7 est impair.
```

2. Au cinéma, la place coûte 6 € pour les moins de 12 ans, 8 € de 12 à 17 ans, 11 € à partir de 18 ans et 7 € à partir de 65 ans. Demande l'âge et affiche le prix.

EXEMPLE D'ÉCRAN

```
Quel âge as-tu ?
70
Ta place coûte 7 €.
```