

Module 1

Alternatives et conditions composées

DA & IA – Bloc 1

Organisation du cours

- Qu'est-ce qu'une alternative?
 - Alternative simple
 - Alternative double
 - Imbrication d'alternatives
- Qu'est-ce qu'une condition?
 - Condition simple
 - Condition composée
- Opérateurs logiques
- Opérateurs logiques et informatique
- Négation d'une condition composée et lois de De Morgan
- Utilisation optimale des booléens
- CASE
- Exercices

Qu'est-ce qu'une alternative?

Instruction → **ACTIONS CONDITIONNELLES**

↪ En fonction de la situation,
il y a différentes façons d'agir.

Permet d'effectuer des séquences d'instructions
différentes selon qu'une situation se présente ou
pas !



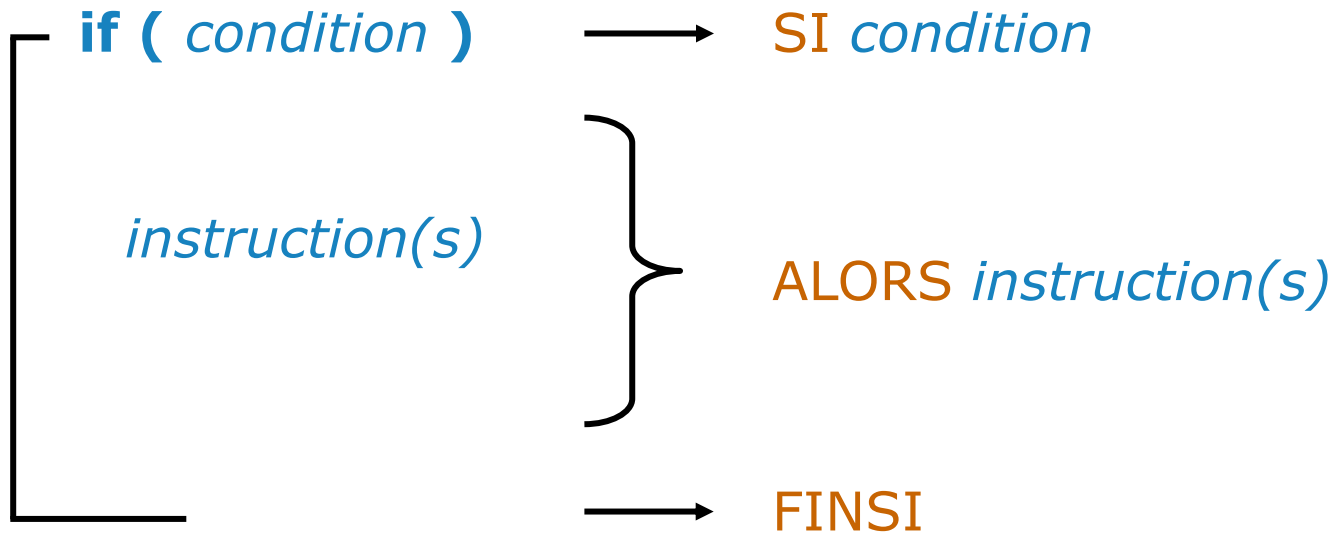
Structure alternative

Alternative

- Il y a plusieurs formes de structures alternatives dont :
 - L'alternative simple
 - L'alternative double
- La structure alternative comporte une condition.

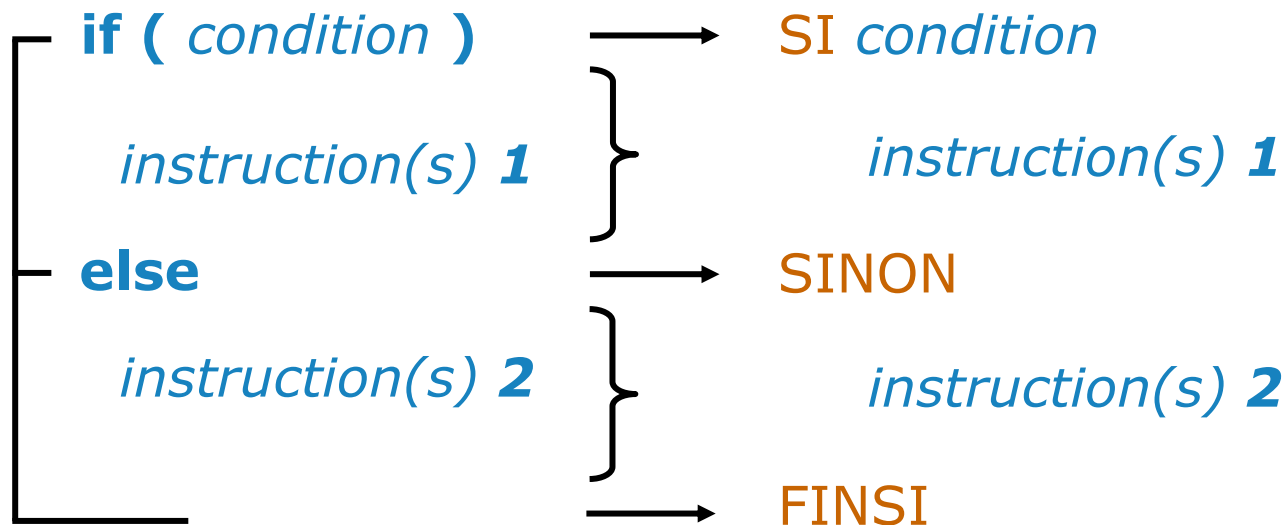
Alternative simple

Le diagramme d'actions :



Alternative double

Le diagramme d'actions :





Imbrication d'alternatives

```
if ( Condition 1 )  
  if ( Condition 2 )  
    instruction(s) 1  
  else  
    instruction(s) 2  
else  
  if ( Condition 3 )  
    instruction(s) 3
```

Qu'est-ce qu'une condition ?

Expression dont la valeur est **VRAI** ou **FAUX**.

C'est un **booléen**.

Condition simple

Cas typique : comparaison entre deux valeurs

↳ Trois éléments :

- une **valeur** (via une variable)
- un **opérateur de comparaison**
- une autre **valeur** (via une variable ou une constante)

Les valeurs peuvent être de **n'importe quel type** :
numériques, alphabétiques ou alphanumériques

💣 Les 2 valeurs doivent être du **même type**



Condition simple

Opérateurs de comparaison

- $==$ ÉGAL À ...
- \neq différent de ...
- $<$ strictement plus petit que ...
- $>$ strictement plus grand que ...
- \leq plus petit ou égal à ...
- \geq plus grand ou égal à ...

Exemples

- $(\text{age} > 18)$
- (trouvé)
- $(\text{nbr1} \leq \text{nbr2})$

Condition



Affirmation vraie ou fausse

Condition composée

C'est une condition composée de plusieurs **conditions simples** unies par un **opérateur logique** (ou connecteur logique).

Exemples

- La valeur de **age** est plus grande que 18 **ET** la valeur de **nom** est plus petite que "Leblanc" :
(age > 18 **AND** nom < « Leblanc »)
- La valeur de **age** est plus grande que 18 **OU** la valeur de **trouvé** est égale à VRAI :
(age > 18 **OR** trouvé)

Opérateurs logiques

AND (\equiv ET)

Pour que la condition composée `cond1 AND cond2` soit VRAI,

il faut que `cond1` soit VRAI **et** que `cond2` soit VRAI.

Table de vérité

cond1	cond2	cond1 AND cond2
VRAI	VRAI	VRAI
VRAI	FAUX	FAUX
FAUX	VRAI	FAUX
FAUX	FAUX	FAUX

Opérateurs logiques

OR (\equiv OU)

Pour que la condition composée **cond1 OR cond2** soit VRAI,

il faut que **cond1** soit VRAI **ou** que **cond2** soit VRAI **ou** que les deux conditions soient VRAI.

Table de vérité

cond1	cond2	cond1 OR cond2
VRAI	VRAI	VRAI
VRAI	FAUX	VRAI
FAUX	VRAI	VRAI
FAUX	FAUX	FAUX

Exercice

- Un centre culturel organise un atelier « Flamenco » dont le tarif annuel est de
 - 100 euros pour les enfants de 6 à 11 ans
 - 120 euros pour les jeunes de 12 à 18 ans
 - 160 euros pour les personnes de plus de 18 ans
- Écrivez le diagramme d'actions qui détermine et affiche le prix à payer par une personne qui s'inscrit à l'atelier et dont on obtient l'âge.

Remarque : il faut 6 ans minimum pour s'inscrire et si ce n'est pas le cas, votre diagramme doit afficher un message le rappelant.
- Critiquez les différentes versions proposées.

Solution (version 1)

```
// version 1
O-----O
| afficherTarifFlamenco |
O-----O

*
obtenir age
┌ if (age < 6)
│   sortir "il faut 6 ans minimum"
└

┌ if(age ≥ 6 AND age ≤ 11)
│   prix = 100
└

┌ if(age ≥ 12 AND ≤ 18)
│   prix = 120
└

┌ if( age ≥ 18)
│   prix = 160
└

┌ if (age ≥ 6)
│   sortir prix
└
```

- Un centre culturel organise un atelier « Flamenco » dont le tarif annuel est de
 - 100 euros pour les enfants de 6 à 11 ans
 - 120 euros pour les jeunes de 12 à 18 ans
 - 160 euros pour les personnes de plus de 18 ans
- Écrivez le diagramme d'actions qui détermine et affiche le prix à payer par une personne qui s'inscrit à l'atelier et dont on obtient l'âge.
Remarque : il faut 6 ans minimum pour s'inscrire et si ce n'est pas le cas, votre diagramme doit afficher un message le rappelant.
- Critiquez les différentes versions proposées

Solution (version 2)

```
// version 2
O-----O
| afficherTarifFlamenco |
O-----O

*
obtenir age
  if (age < 6)
    sortir "il faut 6 ans minimum"
  else
    if(age ≥ 6 AND age ≤ 11)
      prix = 100
    else
      if(age ≥ 12 AND age ≤ 18)
        prix = 120
      else
        if( age ≥ 18)
          prix = 160
    sortir prix
  sortir prix
```

- Un centre culturel organise un atelier « Flamenco » dont le tarif annuel est de
 - 100 euros pour les enfants de 6 à 11 ans
 - 120 euros pour les jeunes de 12 à 18 ans
 - 160 euros pour les personnes de plus de 18 ans
- Écrivez le diagramme d'actions qui détermine et affiche le prix à payer par une personne qui s'inscrit à l'atelier et dont on obtient l'âge.
Remarque : il faut 6 ans minimum pour s'inscrire et si ce n'est pas le cas, votre diagramme doit afficher un message le rappelant.
- Critiquez les différentes versions proposées

Solution (version 3)

```
// version 3
O-----O
| afficherTarifFlamenco |
O-----O

*
obtenir age
  if (age < 6)
    sortir "il faut 6 ans minimum"
  else
    if( age ≤ 11)
      prix = 100
    else
      if(age ≤ 18)
        prix = 120
      else
        prix = 160
    sortir prix
```

- Un centre culturel organise un atelier « Flamenco » dont le tarif annuel est de
 - 100 euros pour les enfants de 6 à 11 ans
 - 120 euros pour les jeunes de 12 à 18 ans
 - 160 euros pour les personnes de plus de 18 ans
- Écrivez le diagramme d'actions qui détermine et affiche le prix à payer par une personne qui s'inscrit à l'atelier et dont on obtient l'âge.
Remarque : il faut 6 ans minimum pour s'inscrire et si ce n'est pas le cas, votre diagramme doit afficher un message le rappelant.
- Critiquez les différentes versions proposées

Opérateurs logiques et informatique

- La quasi-totalité des langages informatiques savent manipuler des valeurs de vérité sous la forme de **booléens** qui s'écrivent généralement **true** et **false**.
- Ils permettent également d'utiliser des **opérations booléennes** (notamment pour écrire les conditions des alternatives et des répétitives).
- Ils permettent aussi d'utiliser les **connecteurs logiques** AND, OR et NOT.
 - AND s'écrit &&
 - OR s'écrit ||
 - NOT s'écrit ! (parfois ~)

Opérateurs logiques et informatique

Mais les connecteurs **&&** et **||** sont des **opérateurs**
« **court circuit** » !

L'ordinateur évalue de gauche à droite mais si une partie de l'évaluation lui suffit, il ne va pas voir plus loin !

Exemple

```
if (x > 5 || y < 2) ...
```

- On vérifie si la valeur de x est > 5.
- Si **ce n'est pas le cas**, on teste la valeur de y.
- Si **c'est le cas**, on conclut immédiatement que la condition est vraie !

Dans une **disjonction**, dès qu'il y a un **VRAI**, le résultat global est **VRAI**.

Opérateurs logiques et informatique

Exemple

```
if (nbArticles != 0 && total/nbArticles > 7) ...
```

- On vérifie si la valeur de nbArticles est différente de 0.
- Si **c'est le cas**, on calcule la division.
- Si **ce n'est pas le cas** (c-à-d si nbArticles vaut 0), on conclut immédiatement que la condition est fausse sans calculer la division !

Conclusion

L'**ordre a de l'importance** : il permet d'éviter une erreur (division par 0) !

Dans une **conjonction**, dès qu'il y a un **FAUX**, le résultat global est **FAUX**.

Opérateurs logiques et informatique

Pour cette raison, nous considérerons aussi **en PP** que les opérateurs **AND** et **OR** sont des **opérateurs court-circuit**.

Vocabulaire

On parle aussi d'évaluation paresseuse (**lazy**).

Exercice 1

À un moment de l'exécution d'un programme, les variables a, b et c contiennent respectivement les valeurs 7, 12 et -5. Pour chacune des conditions suivantes exprimées dans le langage C, indiquez quelles parties seront effectivement évaluées ainsi que le résultat retourné. (On a noté % l'opération modulo.)

a) `a >= 0 && b <= 0`

b) `b <= 0 && a >= 0`

c) `b % a == -c || a > b`

d) `a > b || b % a == -c`

Exercice 2

À un moment de l'exécution d'un programme, les variables `nbProduits`, `prixUnitaire` et `destination` contiennent respectivement les valeurs `10`, `3.5` et `"France"`.

Pour chacune des affectations suivantes, indiquez quelles parties seront effectivement évaluées ainsi que le résultat retourné.

a) `petitColis = nbProduits < 10;`

b) `grosAchat = nbProduits >= 10 || (nbProd * prixUnitaire >= 300);`

c) `livraisonOfferte = destination == "Belgique" || nbProduits > 7;`

d) `colisSimple = nbProduits < 5 && destination == "Belgique";`

e) `paquetRenforcé = nbProduits > 6 && destination == "France";`

Exercice 3

On exécute l'algorithme suivant. Soulignez les parties qui seront effectivement évaluées ou exécutées et donnez la **valeur finale** de chacune des variables.

```
0-----0
| afficherRésultatEnquete |
0-----0

*
nbAvis = 500
nbAvisPositifs = 270
proportionPositifs = nbAvisPositifs / nbAvis
majoritePositifs = proportionPositifs > 0.5

if (nbAvis > 100 AND majoritePositifs)
    résultat = "significativement positif"

if (nbAvis > 100 AND NOT majoritePositifs)
    résultat = "significativement négatif"

if (nbAvis ≤ 100 OR (proportionPositifs > 0.4 AND proportionPositifs < 0.6))
    résultat = "pas de résultat significatif"

sortir resultat
```


Exercice 4

```
0-----0  
| afficherRésultatEnquete |  
0-----0
```

*

```
nbAvis = 500  
nbAvisPositifs = 270  
proportionPositifs = nbAvisPositifs / nbAvis  
majoritePositifs = proportionPositifs > 0.5
```

```
if (nbAvis > 100 AND majoritePositifs )  
  résultat = "significativement positif"
```

```
if (nbAvis > 100 AND NOT majoritePositifs )  
  résultat = "significativement négatif"
```

```
if (nbAvis ≤ 100 OR (proportionPositifs > 0.4 AND proportionPositifs < 0.6))  
  résultat = "pas de résultat significatif"
```

```
sortir resultat
```

Les parenthèses **()** sont facultatives
car le AND l'emporte sur le OR

Réécrire le DA dans une version plus efficace !

Négation d'une condition composée

Exemples introductifs

Que sait-on concernant les variables de la condition lorsqu'on se trouve à l'endroit marqué par la flèche?

```
*  
if (age > 20 AND niveau == "expert")  
else  
    ← age ≤ 20 OR niveau ≠ "expert"
```

```
*  
if (pluie OR temp < 10)  
else  
    ← NOT pluie AND temp ≥ 10
```

Les lois de De Morgan

Trouvées par **Augustus De Morgan**

Comment **nier une disjonction** ?

Illustration en langage courant :

$p \equiv$ ce programme est écrit en C

$q \equiv$ ce programme est écrit en Java

$p \text{ OR } q \equiv$ ce programme est écrit en C ou en Java

$\text{NOT}(p \text{ OR } q) \equiv$ ce programme n'est pas écrit en C ni en Java.
 \equiv il n'est pas écrit en C et il n'est pas écrit en Java
 $\equiv \text{NOT}p \text{ AND } \text{NOT}q$

Les lois de De Morgan

Comment **nier une conjonction** ?

$p \equiv$ ce café est chaud

$q \equiv$ ce café est bon

$p \text{ AND } q \equiv$ ce café est bon et chaud

$\text{NOT}(p \text{ AND } q) \equiv$ il est faux que ce café est bon et chaud.

\equiv il n'est pas bon ou il n'est pas chaud.

$\equiv \text{NOT } p \text{ OR NOT } q$

Exercice 5

Les conditions suivantes sont utilisées dans une structure alternative `if...else...` dont on aimerait intervertir les parties « if » et « else ».
Écrivez les nouvelles conditions à utiliser pour pouvoir effectuer cette transformation en respectant le langage utilisé (pseudo code ou C).

a) `if (age < 18) ...`

b) `if (nbVies > 0 && nbEnnemisRestants == 0) ...`

Exercice 5

c) `if ((age >= 18 || accompagné))...`

d) `if (nbAchats > 5 AND nbAchats ≤ 10)...`

e) `if (tempsRestant > 0 OR NOT aFinExamen)...`

f) `if (NOT arrivéÀLHeure AND NOT aBonneExcuse)...`

Exercice 5

- g) `if (nbAvis < 100 ||
 (propPositif > 0.4 && propPositif < 0.6))...`
- h) `if (clientRégulier &&
 (nbCommandes > 5 || totalCommandes > 750))...`
- i) `if (codePostal == 5000 || dateNaissance >= 20100101)...`
- j) `if (aucunÉchec && moyenne > 16)...`

Exercice 5

k) `if (cote ≥ 10 OR (cote < 10 AND bienVuDuProf))...`

l) `if ((cote > 12 AND NOT aBrossé) OR
 (cote > 10 AND aucuneErreurGrave AND NOT aBrossé))...`

Utilisation optimale des booléens: exercice 6

Réécrivez les pseudo-codes suivants en une seule instruction.

```
// a
*
┌ if (nombre % 2 == 0)
│   pair = true
└ else
  pair = false
```

```
// b
*
┌ if (température ≥ 0)
│   gel = false
└ else
  gel = true
```

Exercice 6

Réécrivez le pseudo-code suivant en une seule instruction.

```
// c
*
  if (age ≥ 18)
    peutEntrer = true
  else
    if (accompagné)
      peutEntrer = true
    else
      peutEntrer = false
```

Exercice 6

Réécrivez le pseudo-code suivant en une seule instruction.

```
// d
*
texteStylé = false
  if (gras)
    texteStylé = true
  if (italique)
    texteStylé = true
```

```
// e
*
texteNormal = true
  if (gras)
    texteNormal = false
  if (italique)
    texteNormal = false
```

Exercice 6

Réécrivez le pseudo-code suivant en une seule instruction.

```
// f
*
  if (gratuit)
    besoinPaielement = false
  else
    if (déjàPayé)
      besoinPaielement = false
    else
      besoinPaielement = true
```

Exercice 6

Réécrivez le pseudo-code suivant en une seule instruction.

```
// g
*
  if (total ≥ 500)
    if (dateInscription > 2020)
      cadeauOffert = true
    else
      if (fraisPort > 60)
        cadeauOffert = true
      else
        cadeauOffert = false
  else
    cadeauOffert = false
```

Exercice 6

Réécrivez le pseudo-code suivant en une seule instruction.

```
//h  
positif =  $x \geq 0$  ? true : false
```

```
//i  
valide =  $\text{age} > 0$  ?  $\text{age} \leq 18$  : false
```

CASE

Ce type de structure s'utilise lorsque l'alternative porte sur **une seule variable**, de type **numérique** ou **caractère**, et que les tests sont des **égalités**.

Exemple

```
if (choix == 1)
  sortir "Sauvegarde en cours"
else
  if (choix == 2)
    sortir "Fermeture en cours"
  else
    if (choix == 3)
      sortir "Retour"
    else
      sortir "Erreur !"
```



```
if (choix == 1)
  sortir "Sauvegarde en cours"
if (choix == 2)
  sortir "Fermeture en cours"
if (choix == 3)
  sortir "Retour"
else
  sortir "Erreur !"
```

Dès qu'une condition est vérifiée, les instructions sous la condition sont exécutées mais le reste de la structure ne sera pas lu.