



compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
iccompagneme
tertiaire
fication métier
professionnel
compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification



Algorithmique Partie 1

Les instructions de base

12/10/2015

1



compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
iccompagneme
tertiaire
fication métier
professionnel
compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification


Introduction

⇒ Qu'est ce que l'algorithmique

- **Définition:**
C'est l'étude de la résolution de problèmes par la mise en œuvre de suites d'opérations élémentaires selon un processus défini aboutissant à une solution.
(source Wiktionary.org)
- **Remarque:**
Souvent cette solution n'est pas unique.

12/10/2015

2



compétences
bâtimentinserti
rmationterti
erviceemploi accueil
orientation
industrie develop
certification
accompagnement
formation
professionnel
compétences
bâtimentinserti
rmationterti
erviceemploi accueil
orientation
industrie develop
certification


Introduction

⇒ Pourquoi faire de l' « algo » ?

- **Objectif:**
Comment arriver à partir d'un problème réel à un programme pouvant être exécuté par une machine ou un ordinateur.
- **Problème:**
Expliquer pas à pas à l'ordinateur comment il doit si prendre avec ses connaissances pour résoudre le problème posé.

12/10/2015

3



compétences
bâtimentinserti
rmationterti
erviceemploi accueil
orientation
industrie develop
certification
accompagnement
formation
professionnel
compétences
bâtimentinserti
rmationterti
erviceemploi accueil
orientation
industrie develop
certification

Introduction

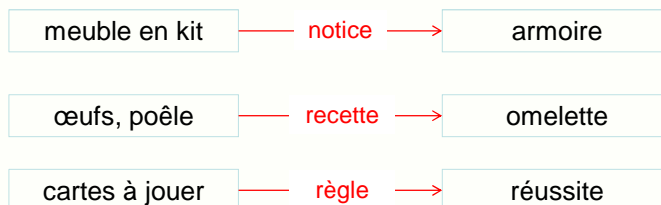
⇒ But du cours d'algorithmique

- Résoudre des problèmes en se mettant à la place de la machine.
- Savoir expliquer son raisonnement.
- Savoir formaliser son raisonnement.
- Concevoir des algorithmes c'est :
 - décrire une succession d'opérations qui, si elles sont correctement exécutées par l'ordinateur, produiront le résultat déterminé
 - un algorithme **n'est pas une solution unique** : pour réaliser une même activité, **plusieurs algorithmes sont possibles.**

12/10/2015

4

- La notion d'algorithme **n'est pas spécifique à l'informatique.**



- En règle générale, l'entité qui exécute une tâche est appelée **processeur.**
- Un processeur obéit aux actions (les exécute) que l'algorithme lui décrit.

12/10/2015

5

- La **complexité** :
Capacité d'un algorithme à traiter des problèmes plus ou moins complexe.
Quel ressources (espace mémoire ...) a-t-il besoin.
- La **rapidité** :
En combien de temps un algorithme va-t-il donner un résultat.

12/10/2015

6



compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
iccompagneme
tertiaire
fication métier
professionnel
compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification

12/10/2015

7

Définition

⇒ Représentation des algorithmes

- Au cours du temps la représentation d'un algorithme a évolué :
 - L'organigramme
 - Les graphes de Nassi-Schneidermann
 - Les arbres programmatiques
 - Le pseudo langage
- Aujourd'hui c'est ce dernier qui est utilisé.



compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
iccompagneme
tertiaire
fication métier
professionnel
compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification

12/10/2015

8

Le pseudo-langage

⇒ Introduction

- Il est aussi appelé **pseudo-code**.
- Il permet de **présenter un algorithme** avec des **instructions évoluées** utilisant des mots en **langage quasi-naturel**.
- Il est une **passerelle** entre **les données de l'analyse** et **les détails du codage**.
- Il offre une possibilité supplémentaire de **documentation de l'application développée**.

⇒ Un premier algorithme

```
// algorithme calculant le périmètre d'un carré
```

```
// préparation du traitement
```

```
// traitement
```

```
// présentation du résultat
```

fin

9

⇒ Les 4 phases d'un algorithme

- 10



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
icompagneme
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

12/10/2015

11

Le pseudo-langage

⇒ Les commentaires

- Pour des raisons de clarté, il est judicieux de mettre **des commentaires**.
- Syntaxe :
 - ceux débutant par # et se finissant par #
#ceci est un commentaire#
 - ceux débutant par { et se finissant par }
{ceci est encore un commentaire}
 - ceux dérivés des langages de programmation:
/ un commentaire
sur plusieurs lignes*/
// suite de la ligne en commentaire*



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
icompagneme
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

12/10/2015

12

Le pseudo-langage

⇒ Les variables : définition

- Une **variable informatique** est une entité qui permet à la machine de stocker une valeur.
- Une variable sera **identifiée** par un **nom** unique.
- La valeur de la variable peut évoluer au cours de l'exécution de l'algorithme.
- Une variable est **typée** ce qui détermine l'ensemble des valeurs qu'elle peut prendre.



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
accompagnement
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

Le pseudo-langage

⇒ Les variables : déclaration

- Déclarer une variable :

variable <liste d'identificateurs> : type

Une variable doit être déclarée avant son utilisation.

- L'identificateur :

On utilisera la notation « camel case » le 1^{er} mot en minuscule les autres la première lettre en majuscule

Exemples :

variable nbrArticle : Entier

delta, resultatHorsTaxe : Réel

12/10/2015

13



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
accompagnement
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

Le pseudo-langage

⇒ Les variables : les types

- Les variables sont **typées** en fonction du genre de contenu qu'elles représentent.
- Les types courants sont:

Entier : entiers positif ou négatif
Réel, Double : décimaux
Caractère : lettre, chiffre, symbole
Chaîne de caractères
Booléen : vrai, faux

- Si plusieurs variables sont du même type, on les déclare en les séparant par une virgule.

12/10/2015

14



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
accompagnement
formation
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

12/10/2015

15

Le pseudo-langage

⇒ L'instruction d'affectation

- L'affectation permet de **donner** ou de **modifier** la valeur d'une variable (L'ancienne valeur est **écrasée** par la nouvelle).
- La syntaxe est une flèche allant de la droite vers la gauche :

variable ← valeur

- Valeur peut être le résultat d'un calcul ou une autre variable du même type

Exemple :

circonf ← 2*3.14*rayon

- On peut se rapprocher de la syntaxe des langages informatiques :

nombrePI = 3.14



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
accompagnement
formation
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

12/10/2015

16

Le pseudo-langage

⇒ L'instruction de saisie

- Il est intéressant de demander à l'utilisateur de rentrer des valeurs au clavier qui seront placer dans des variables.
- Les mots clés utilisés sont « saisir, entrer »
- Ils seront utilisés sous forme de « fonction » (utilisation des parenthèses)

Exemple :

saisir(rayon)
entrer(nom, prenom)



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
accompagnement
formation
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

Le pseudo-langage

⇒ L'instruction d'affichage

- Un algorithme étant fait pour exécuter un traitement, il est donc nécessaire d'afficher les résultats.
- Le mot utilisé est « afficher »

Exemple :

```
afficher("la circonférence est :", circonfer)  
afficher("la somme des nombre est :",  
        nbr1 + nbr2)
```

12/10/2015

17



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
accompagnement
formation
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

Le pseudo-langage

⇒ Les constantes

- Les constantes permettent de donner un nom à des valeurs fixées une fois pour toute.
- Elles apportent de la lisibilité au code.
- Le nom sera de préférence en **majuscule** pour ne pas les confondre avec les variables (utiliser under score pour séparer les mots).
- Elles sont déclarées comme les variables mais avec le mot clé **constante**.

```
constante (nom : type) ← valeur
```

Exemple :

```
constante (NBR_MAX : Entier) ← 15
```

12/10/2015

18



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
accompagnement
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

12/10/2015

19

Le pseudo-langage

⇒ La séquence

- **Définition :**
Suite d'instructions s'exécutant les unes après les autres dans l'ordre d'écriture
- **Identification :**
Pour identifier une séquence on encadre les instructions par les mots « **début** » et « **fin** ».
- **Exemple :**

```
constante PI ← 3.14  
variable rayon, circonf : réel  
début  
  saisir(rayon)  
  circonf ← 2 x PI x rayon  
  afficher( "La circonférence est : ",circonf)  
fin
```



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
accompagnement
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

12/10/2015

20

Le pseudo-langage

⇒ La sélection

Deux types de construction permettent de choisir (ou sélectionner) une séquence d'instructions en fonction de conditions précises.

- La construction **si-alors-sinon**
(dite « *alternative* »)
La condition est binaire
- La construction **selon**
(dite « *choix multiple* »)
La sélection se fait sur des valeurs discrètes prises par une variable



compétences
bâtiment inserti
rmation terti
ervice emploi accuei
orientation
industrie develop
certification
icompagneme
tertiaire
fication
professionnel
compétences
bâtiment inserti
rmation terti
ervice emploi accuei
orientation
industrie develop
certification

12/10/2015

21

Le pseudo-langage

⇒ La sélection : SI-ALORS-SINON

▪ Format général :

```
si ( condition binaire ) alors  
    < séquence d'instructions vraies >  
Sinon  
    < séquence d'instructions fausses >  
finSi
```

- La condition testée est une **condition logique BINAIRE**.
- Elle admet que 2 solutions : elle est **vraie** ou elle est **fausse**.



compétences
bâtiment inserti
rmation terti
ervice emploi accuei
orientation
industrie develop
certification
icompagneme
tertiaire
fication
professionnel
compétences
bâtiment inserti
rmation terti
ervice emploi accuei
orientation
industrie develop
certification

12/10/2015

22

Le pseudo-langage

⇒ La sélection : SI-ALORS-SINON

Exemple :

```
si ( X >= 0 )    // X est une variable quelconque  
    alors  
        afficher(" X est positif ou nul ")  
    sinon  
        afficher(" X est négatif ")  
finSi
```

- Les instructions derrière le mot-clé **ALORS** sont appelées la clause « **VRAI** ».
- Les instructions derrière le mot-clé **SINON** sont appelées la clause « **FAUX** ».



Le pseudo-langage

⇒ La sélection : SI-ALORS-SINON

- La clause SINON peut être vide :

si (c'est le week-end) **alors**

début// Séquence

Je tonds la pelouse

Je lave la voiture

fin

finSi

- Si ce n'est pas le week-end, je ne fais rien
- Remarque: les mot clés « début » « fin » peuvent être omis

12/10/2015

23



Le pseudo-langage

⇒ La sélection : SI-ALORS-SINON

- L'imbrication est possible :

si (c'est le week-end) **alors**

si(il ne pleut pas) **alors**

début// Séquence

Je tonds la pelouse

Je lave la voiture

fin

sinon

je joue aux cartes

finSi

finSi

Respecter l'indentation
pour la clarté et la
lisibilité du code

12/10/2015

24



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
accompagnement
formation
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

12/10/2015

25

Le pseudo-langage

⇒ La sélection : SI-ALORS-SINON

- La condition binaire peut être complexe :

```
si((c'est le week-end) et (il fait beau)) alors  
  début// Séquence  
    Je tonds la pelouse  
    Je lave la voiture  
  fin  
finSi
```

- La négation : opérateur « NON » ou « ! »
- Le ET logique : opérateur « ET »
- Le OU logique : opérateur « OU »



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
accompagnement
formation
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

12/10/2015

26

Le pseudo-langage

⇒ La sélection : SELON

- Format général :

```
selon ( sélecteur )  
  val1 :      instructions pour sélecteur= val1  
  val2, val3 : instructions pour val2 et val3  
  val4 :      instructions pour val4  
  autres :    instructions pour autres valeurs  
finSelon
```

- Le « sélecteur » (qui est souvent une variable) peut prendre les valeurs **énumérées** par les val1 à valn



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
accompagnement
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

Le pseudo-langage

⇒ La sélection : SELON

- Exemple :

selon (ceJour)

"Samedi" : Tondre la pelouse

"Dimanche" : Jouer aux cartes

autres : Se rendre au travail

finSelon

- Samedi et Dimanche sont des valeurs possibles pour le sélecteur « ce_jour »

12/10/2015

27



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
accompagnement
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

Le pseudo-langage

⇒ La sélection : SELON

- Les valeurs du sélecteur peuvent aussi être regroupées

selon(ceJour)

"Lundi", "Mardi" : je fais mon travail de début de semaine

"Mercredi", "Jeudi" : je fais mon travail de milieu de semaine

"Vendredi", "Samedi" : je fais mon travail de fin de semaine

finSelon

12/10/2015

28



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
icompagneme
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

12/10/2015

29

Le pseudo-langage

⇒ L'itération

- Pour répéter une même séquence un certain nombre de fois , **plusieurs constructions sont possibles.**
 - La construction **TANTQUE-FAIRE**
La séquence se fait tant qu'une condition logique est vraie
 - La construction **REPETER-JUSQUE**
La séquence se fait jusqu'à ce qu'une condition logique soit vraie
 - La construction **POUR_FAIRE**
La séquence se fait un nombre déterminé de fois



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification
icompagneme
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
develop
certification

12/10/2015

30

Le pseudo-langage

⇒ L'itération : TANTQUE-FAIRE

- Format général :

Initialisation de la condition
tantque (condition) faire
séquence
réévaluation de la condition
finTantque

1. Avec cette construction, la condition est d'abord évaluée.
2. Ensuite, si la condition est vraie, la séquence suivant le mot-clé FAIRE est ré exécutée.



compétences
bâtiment
inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
icompagneme
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification

Le pseudo-langage

⇒ L'itération : TANTQUE-FAIRE

Exemple :

```
LaTemperature = LireLaTempérature()  
tantque(LaTemperature >= 50° C )  
  faire  
    début  
      Maintenir l'air conditionné  
      LaTemperature ← LireLaTempérature()  
    fin  
finTantque
```

- Remarque 1 : LireLaTempérature()
- Remarque 2 : DEBUT, FIN

12/10/2015

31



compétences
bâtiment
inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
icompagneme
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification

Le pseudo-langage

⇒ L'itération : TANTQUE-FAIRE

- Quelques points à considérer :
 - le **corps** de la boucle est une **séquence d'instructions**.
 - la condition doit être **binaire** et peut être **complexe** (usage d'opérateurs booléens).
 - la condition est évaluée **en tout premier** => il est **possible** que le corps de boucle ne soit **jamais exécuté**.
 - la **création d'une boucle infinie est possible** : (la condition est toujours vraie)

12/10/2015

32



Le pseudo-langage

⇒ L'itération : REPETER-TANTQUE

■ Format général :

```
répéter  
    séquence  
        réévaluer la condition  
tantque( condition )
```

1. Avec cette construction, la séquence s'effectue
2. La condition doit être évaluée avant la fin de la boucle.
3. Si la condition est vraie la séquence est ré exécutée.

12/10/2015

33



LE PSEUDO-LANGAGE

L'itération : REPETER-JUSQUE

Exemple :

```
répéter  
    début  
        Maintenir l'air conditionné  
        LaTemperature ← LireLaTempérature()  
    fin  
tantque( LaTemperature > 50°C )
```

- Remarque 1 : LireLaTempérature()
- Remarque 2 : DEBUT, FIN

12/10/2015

34



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
accompagnement
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

12/10/2015

35

LE PSEUDO-LANGAGE

L'itération : TANTQUE-FAIRE

- Quelques points à considérer :
 - le **corps** de la boucle est une **séquence d'instructions**.
 - la condition doit être **binaire** et peut être **complexe** (usage d'opérateurs booléens).
 - La condition est évaluée **en sortie de boucle**
=> **le corps de boucle est exécutée au moins une fois.**
 - la **création d'une boucle infinie est possible** : (la condition est toujours vraie)



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
accompagnement
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

12/10/2015

36

Le pseudo-langage

⇒ L'itération : POUR-FAIRE

- Format général :

pour varIndex ← valInit **à** valFin **faire**
séquence
finPour
- La variable « varIndex » prend les valeurs **numériques** de valInit à valFin.
- Elle s'incrémente de 1 à chaque tour de boucle.
- On peut ajouter l'option **par pas de x** avant le mot clé **faire**.
- Avec cette construction, la séquence s'effectue **un nombre de fois déterminé**.



compétences
bâtiment
information
service emploi accueil
orientation
industrie develop
certification
accompagnement
formation
professionnel
compétences
bâtiment
information
service emploi accueil
orientation
industrie develop
certification

Le pseudo-langage

⇒ L'itération : POUR-FAIRE

Exemple :

```
pour compteur ← 1 à 10 faire  
  début  
    résultat = compteur * 5  
    Afficher ( résultat )  
  fin  
finPour
```

- Remarque 1 : compteur
- Remarque 2 : DEBUT, FIN

12/10/2015

37



compétences
bâtiment
information
service emploi accueil
orientation
industrie develop
certification
accompagnement
formation
professionnel
compétences
bâtiment
information
service emploi accueil
orientation
industrie develop
certification


Le pseudo-langage

⇒ L'itération : POUR-FAIRE

- Il faut noter que :
 - L'initialisation est faite au début de la boucle.
 - La valeur située après « à » représente une condition de fin de boucle.
 - L'incrément est en général égal à l'unité, il peut être adjoint l'option « **par pas de n** » pour changer la valeur de l'incrément qui peut être positif ou négatif.
 - La variable de boucle ne doit pas être modifiée à l'intérieur de la boucle.

12/10/2015

38



compétences
bâtiment
inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
iccompagneme
tertiaire
fication
professionnel
compétences
bâtiment
inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification

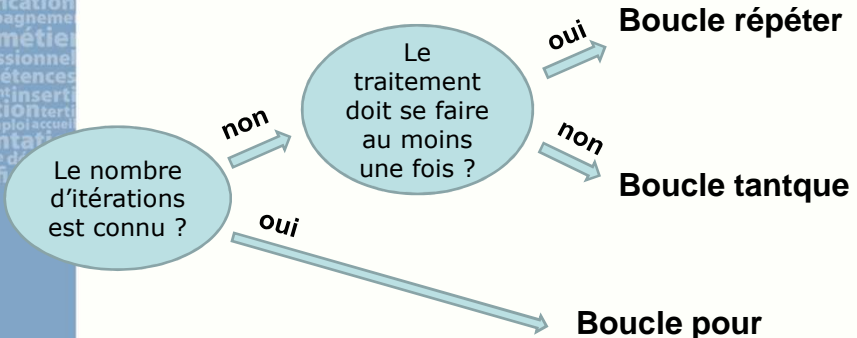
12/10/2015

39

Le pseudo-langage


⇒ L'itération : synthèse

- Choisir pour ... tantque ... répéter ... ?



```

graph TD
    A([Le nombre d'itérations est connu ?]) -- oui --> C[Boucle pour]
    A -- non --> B([Le traitement doit se faire au moins une fois ?])
    B -- oui --> D[Boucle répéter]
    B -- non --> E[Boucle tantque]
  
```



compétences
bâtiment
inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
iccompagneme
tertiaire
fication
professionnel
compétences
bâtiment
inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification

12/10/2015

40

Le pseudo-langage

⇒ Exemple : préparation d'une omelette

- Enoncé : Préparer une omelette
 - **Phase A : Casser** les œufs dans un bol.
 - **Phase B : Battre** les blancs et les jaunes avec une fourchette.
 - **Phase C : Mettre** de l'huile à chauffer dans une poêle sur la cuisinière.
 - **Phase D** : Lorsque la poêle est chaude, y **verser** le contenu du récipient.
 - **Phase E : Enlever** la poêle de la cuisinière quand l'omelette est cuite.
 - **Phase F : Eteindre** le feu.



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
icompagneme
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

12/10/2015

41

Le pseudo-langage

⇒ Exemple : préparation d'une omelette

■ Phase A :

A1 : Poser les œufs sur le plan de travail

répéter

début

A2 : Prendre un œuf sur le plan de travail

A3 : Le casser et verser son contenu dans le bol

A4 : Jeter la coquille dans la poubelle

fin

tantque(*il y a des œufs sur le plan de travail*)

REMARQUE IMPORTANTE :

- Les autres phases (B,C,D,E,F) qui ne peuvent pas être décomposées sont appelées **actions primitives**.



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
icompagneme
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

12/10/2015

42

Le pseudo-langage

⇒ Remarques

- finSi, finTantque et finPour peuvent être omis si le corps (séquence) se limite à une seule instruction.

si (val > 0) alors afficher("fini!")

- Les délimiteur de séquence « début » « fin » peuvent être omis si la séquence comporte plusieurs instructions.

pour compteur ← 1 à 10 **faire**

résultat = compteur * 5

afficher (résultat)

finPour



compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
iccompagneme
tertiaire
fication
professionnel
compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification

12/10/2015

43

La logique propositionnelle

⇒ Introduction

- La logique : une façon de formaliser notre raisonnement
 - La **logique propositionnelle**: modèle mathématique qui nous permet de raisonner sur la nature vraie ou fausse des expressions logiques.
 - En logique classique ([logique bivalente](#)), une proposition peut prendre uniquement les valeurs *vrai* ou *faux*.
 - un prédicat est lui une proposition dont la valeur de vérité dépend de variables qu'elle renferme.



compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification
iccompagneme
tertiaire
fication
professionnel
compétences
bâtiment inserti
rmation terti
ervice emploi accueil
orientation
industrie develop
certification

12/10/2015

44

La logique propositionnelle

⇒ les conditions de test et de boucle

Regardons les expressions se trouvant dans les conditions de test ou de boucle :

- si température \leq SEUIL alors...*
- tantque saisie \neq STOP et nbrVal $<$ MAX faire ...*
- tantque valeur < 0 ou (valeur % 2) $\neq 0$ faire...*
- Les expressions peuvent prendre la valeur **VRAI** ou **FAUX** en fonction des valeurs des variables
- Elles peuvent être complexes



La logique propositionnelle

⇒ les éléments

- Formule :
 - expression logique composée de variables propositionnelles et de connecteurs logiques
- Variable propositionnelle :
 - une proposition considérée comme indécomposable
- Connecteurs logiques:
 - Négation **non** (\neg) conjonction **et** (\wedge)
 - disjonction **ou** (\vee)

Exemple : a et b variables propositionnelles

$((A \text{ ou } B) \text{ et non} B) \text{ ou } (A \text{ ou non} B)$

12/10/2015

45



La logique propositionnelle

⇒ tables de vérité

- Représentation des tables de vérité associées à une expression logique

négation

A	non A
V	F
F	V

conjonction

A	B	A et B
F	F	F
F	V	F
V	F	F
V	V	V

disjonction

A	B	A ou B
F	F	F
F	V	V
V	F	V
V	V	V

12/10/2015

46



La logique propositionnelle

⇒ équivalences

- Commutativité
 - A et B équivalent à B et A
 - A ou B équivalent à B ou A
- Associativité
 - A et (B et C) équivalent à (A et B) et C
 - A ou (B ou C) équivalent à (A ou B) ou C
- Distributivité
 - A et (B ou C) équivalent à (A et B) ou (A et C)
 - A ou (B et C) équivalent à (A ou B) et (A ou C)

12/10/2015

47



La logique propositionnelle

⇒ remarques

- Toujours VRAI
 - A ou (non A) équivalent à VRAI
- Toujours FAUX
 - A et (non A) équivalent à FAUX

Exemple :

être mineur (*m*) ou
(majeur (*non m*) non imposable (*non i*))
équivalent à
être mineur (*m*) ou non imposable (*non i*) "
 $m \text{ ou } ((\text{non } m) \text{ et } (\text{non } i)) \Leftrightarrow m \text{ ou } (\text{non } i)$

12/10/2015

48

Lois de Morgan

- non (A et B) équivalent à (non A) ou (non B)
- non(A ou B) équivalent à (non A) et (non B)

Exemple :

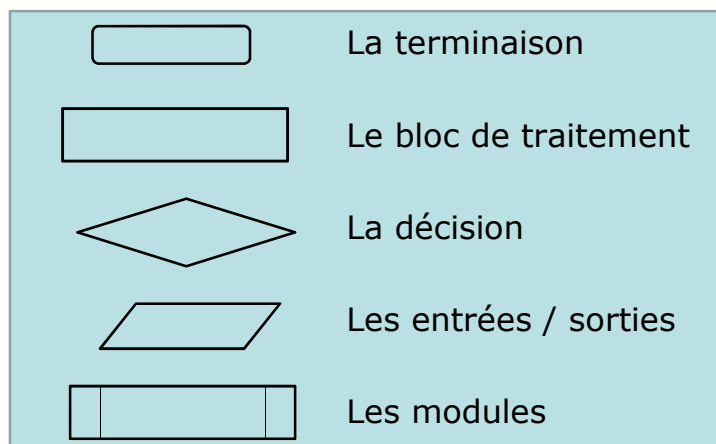
```
si(saisie <> FIN et nbrVal < 10) alors
    enregistrer saisie
sinon
    afficher("fin de saisie")
finSi
```

```
si(saisie = FIN ou nbrVal >= 10) alors
    afficher("fin de saisie")
sinon
    enregistrer saisie
finSi
```

12/10/2015

49

Les principaux symboles :



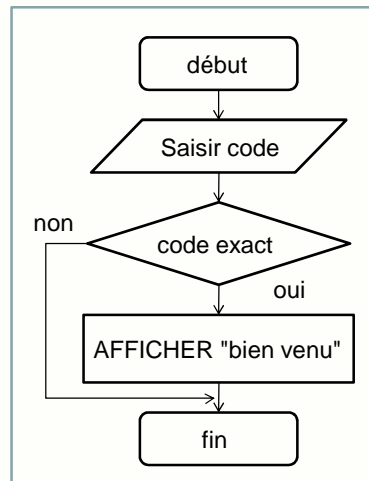
12/10/2015

50

Exemple :

```

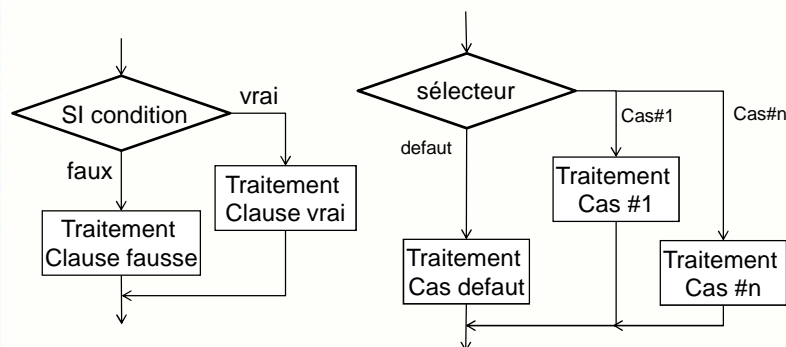
début
constante (PSW : chaîne) ← "moi"
variable code : chaîne
saisir( code )
si (code = PSW) alors
    afficher("bien venu")
finSi
fin
    
```



12/10/2015

51

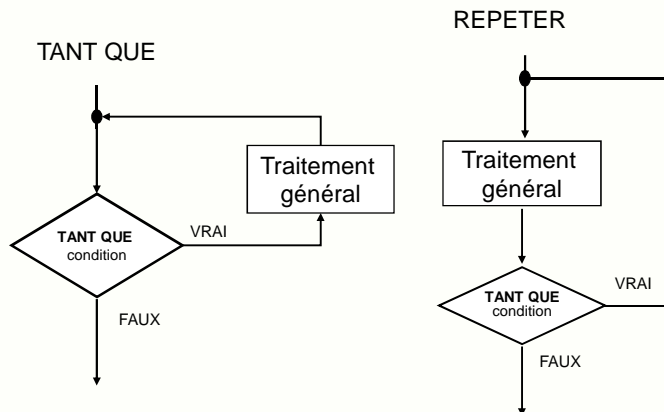
▪ Les sélections :



12/10/2015

52

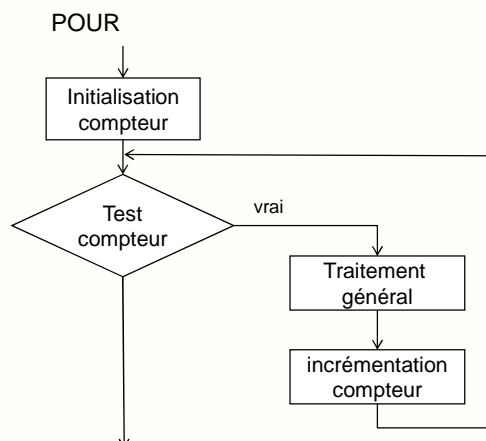
■ Les itérations :



12/10/2015

53

■ Les itérations :



12/10/2015

54



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
icompagneme
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

12/10/2015

55

Autres représentations

⇒ les graphes de Nassi-Schneidermann

■ Introduction :

- [Isaac Nassi](#) et [Ben Shneiderman](#) ont développé cette représentation en 1972
- Représentation graphique basée sur des diagrammes emboîtés.
- Encore connu sous les noms de NSD ou de structogramme.



compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification
icompagneme
tertiaire
fication
métier
professionnel
compétences
bâtiment
inserti
rmation
terti
ervice
emploi
accueil
orientation
industrie
dévelop
certification

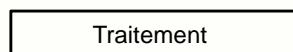
12/10/2015

56

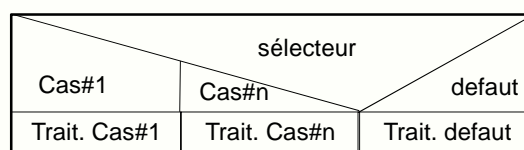
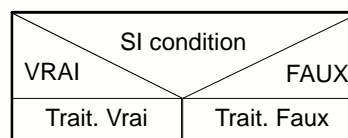
Autres représentations

⇒ les graphes de Nassi-Schneidermann

■ La séquence :



■ La sélection :

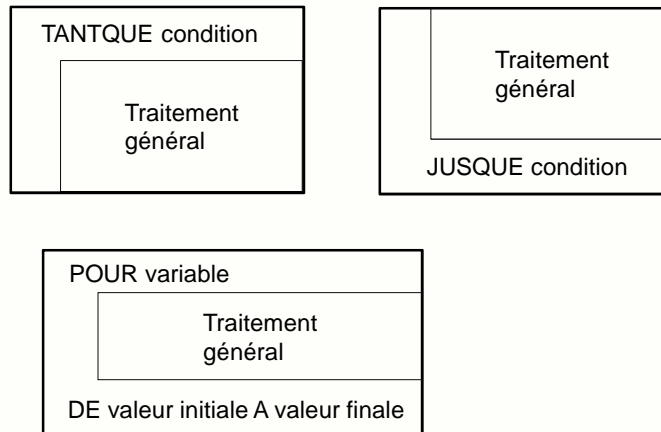




Autres représentations

⇒ les graphes de Nassi-Schneidermann

■ Les itérations :



12/10/2015

57



Autres représentations

⇒ les arbres programmatiques

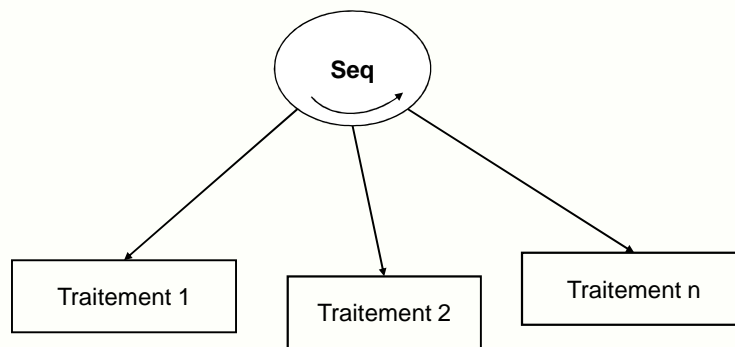
■ Introduction :

- Les arbres programmatiques sont issus de la théorie des arbres.
- C'est une représentation graphique où la lecture se fait en tournant dans le sens trigonométrique.

12/10/2015

58

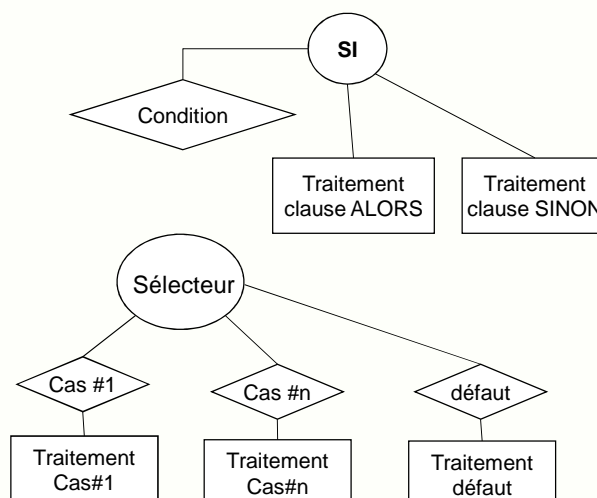
La séquence :



12/10/2015

59

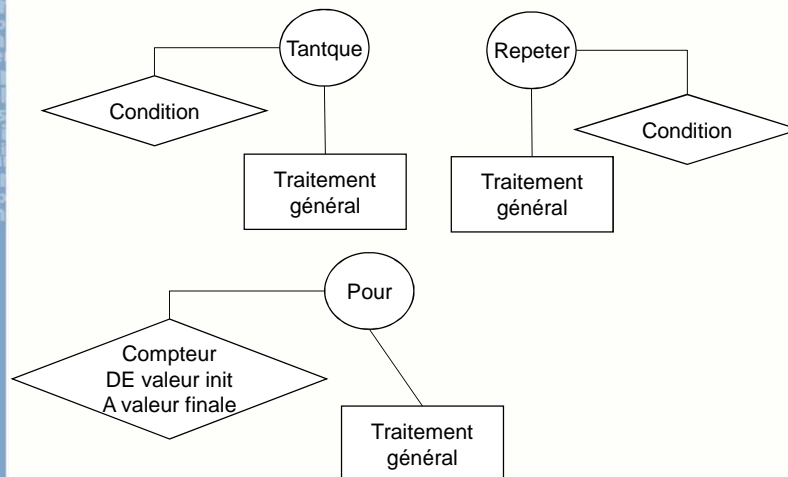
La sélection :



12/10/2015

60

■ Les itérations :



12/10/2015

61

Fin de la première partie

12/10/2015

62