

GENERALITES

Michael
X²NATIS



Compétence demandée :
Comprendre les mécanismes algorithmiques

1. L'algorithmique et les maths ?
2. Enjeux de l'informatique
3. Types et structures
4. Algorithme
5. Contrôle du flux
6. Organigramme

1. L'algorithmique et les maths ?
2. Enjeux de l'informatique
3. Types et structures
4. Algorithme
5. Contrôle du flux
6. Organigramme

L'algorithmique et les maths

L'algorithmique est l'étude et la production de règles et techniques qui sont impliquées dans la définition et la conception d'algorithmes, c'est-à-dire de processus systematiques de résolution d'un problème permettant de décrire précisément des étapes pour résoudre un problème.

Les mathématiques (ou la mathématique) sont un ensemble de connaissances abstraites résultant de raisonnements logiques appliqués à des objets divers tels que les ensembles mathématiques, les nombres, les formes, les structures, les transformations, etc. ; ainsi qu'aux relations et opérations mathématiques qui existent entre ces objets.

Il faut comprendre 3 choses :

- Un carré ne rente pas dans un cercle et un cercle ne rentre pas dans un carré
- L'informatique ne se limite pas à l'étude des nombres
- Une fonction produit un résultat d'une forme particulière

1. L'algorithmique et les maths ?
2. Enjeux de l'informatique
3. Types et structures
4. Algorithme
5. Contrôle du flux
6. Organigramme

Enjeux de l'informatique

Un programme informatique est un **ensemble d'instructions et d'opérations** destinées à être exécutées par un ordinateur.

Un programme fait généralement partie d'un logiciel que l'on peut définir comme un ensemble de composants numériques destiné à fournir un service informatique

Un **code source** est un code écrit par un informaticien dans un langage de programmation. Il peut être compilé vers **une forme binaire** ou directement interprété.

Une **machine de Turing** est un **modèle abstrait du fonctionnement** des appareils mécaniques de calcul, tel un **ordinateur**. Ce modèle a été imaginé par Alan Turing en 1936, en vue de donner une définition précise au concept d'algorithme ou de « procédure mécanique ».

La programmation repose sur une hypothèse fondamentale : **l'universalité de Turing**. En effet, Alan Turing a fait de l'informatique une science en créant un modèle d'ordinateur universel, la machine de Turing. **Tout ce qui peut être calculé par un algorithme le peut par cette machine.**

Les algorithmes donnent naissance à des programmes qui s'exécutent sur la machine.

Un programme :

- Prend du temps
- Prend de l'espace

On parle d'exécution

La **théorie de la complexité** est le domaine des mathématiques, et plus précisément de l'informatique théorique, qui étudie formellement d'abord **le temps de calcul**, mais aussi **l'espace mémoire** (et plus marginalement la taille d'un circuit, le nombre de processeurs, ...) requis par un algorithme pour résoudre un problème algorithmique

1. L'algorithmique et les maths ?
2. Enjeux de l'informatique
3. Types et structures
4. Algorithme
5. Contrôle du flux
6. Organigramme

Types et structures

Les **types primitifs** constituent un **format** (une convention) de données que l'ordinateur peut traiter de **manière unitaire**.

Un **type primitif** décrit l'**agencement particulier des bits dans l'ordinateur**. Le plus ancien type énoncé fut le word en langage B.

Un **tableau** (tableau) est une **structure de données** contenant un groupe d'éléments **tous du même type**, avec **des adresses consécutives** sur la mémoire (memory). Le tableau a le **nombre fixé d'éléments** et vous ne pouvez pas changer sa taille.

Une **classe**, hors cadre de la POO, est un ensemble personnalisé de types existants décrivant un agencement de données. Elle permet donc de décrire une distribution des types dans un espace mémoire.

1. L'algorithmique et les maths ?
2. Enjeux de l'informatique
3. Types et structures
4. **Algorithme**
5. Contrôle du flux
6. Organigramme

Algorithme

Un **algorithme** est une **suite finie et non ambiguë** d'instructions et d'opérations permettant de résoudre une classe de problèmes.

Le domaine qui étudie les algorithmes est appelé l'**algorithmique**.

On retrouve aujourd'hui des algorithmes dans de nombreuses applications telles que le fonctionnement des ordinateurs, la cryptographie, le routage d'informations, la planification et l'utilisation optimale des ressources, le traitement d'images, le traitement de textes, la bio-informatique, etc

Les algorithmes sont construits à partir de séquences, sélection et répétition.

Aujourd'hui, nous disons que les algorithmes sont construits à partir de variables, blocs, conditions et de boucles.

Une **procédure** est une instruction composée qui peut prendre des **paramètres** et dont le rôle est de modifier l'état courant. Les procédures ne retournent pas de résultat.

Une fonction renvoie une valeur et une procédure exécute uniquement des commandes.

La fonction de nom vient des maths. Il est utilisé pour calculer une valeur en fonction des entrées qui pourra être à la suite utilisée dans une autre fonction ou dans une autre procédure.

1. L'algorithmique et les maths ?
2. Enjeux de l'informatique
3. Types et structures
4. Algorithme
5. Contrôle du flux
6. Organigramme

Contrôle du flux

Le flux d'un programme ?

Une **structure de contrôle** est une instruction particulière d'un langage de programmation impératif pouvant **dévier le flux** de contrôle du programme la contenant lorsqu'elle est exécutée

Si, au plus bas niveau, l'éventail se limite généralement aux **branchements et aux appels de sous-programme**, les langages structurés offrent des constructions plus élaborées comme les alternatives (if, if-else, switch...), les boucles (while, do-while, for...) ou encore les appels de fonction.

1. L'algorithmique et les maths ?
2. Enjeux de l'informatique
3. Types et structures
4. Algorithme
5. Contrôle du flux
6. Organigramme

Organigramme



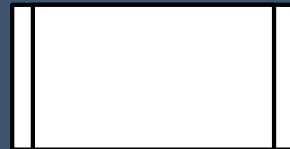
Début, fin

Début ou fin d'un programme



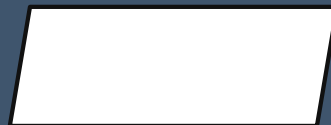
Action, traitement

Opération ou calcul sur des données dont le résultat reste dans la machine.



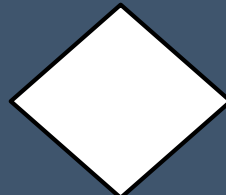
Macro, sous-programme

Appel d'une portion de programme considérée comme une simple opération.



Entrée, sortie

Lecture ou écriture d'une information, sur un port d'entrée sortie.



Décision

Test, décision, exploitation de conditions variables impliquant le choix d'une parmi deux.

Répétition

Faire

Tant que

Tant que

Fin tant que

Pour (Initialisation, Condition, Pas)

Fin pour

Décision

Si

Alors

Sinon

Fin si

Choix

Cas 1 :

Cas n :

Défaut :

Fin choix

Si

Alors

Fin si

Autre

Début

Attendre

Fin

Constante

Variables

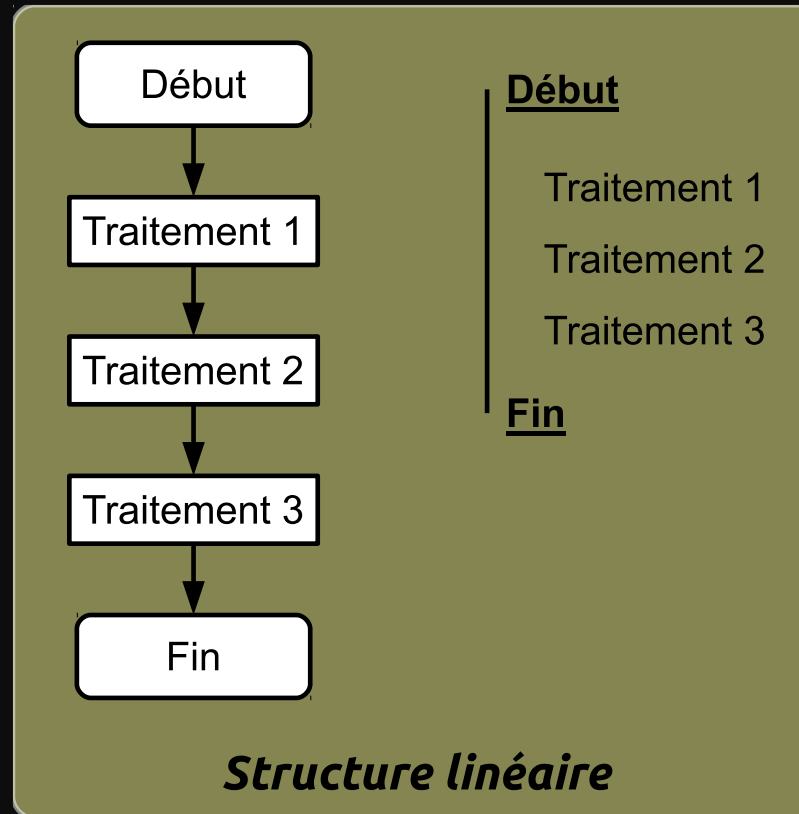
Entier

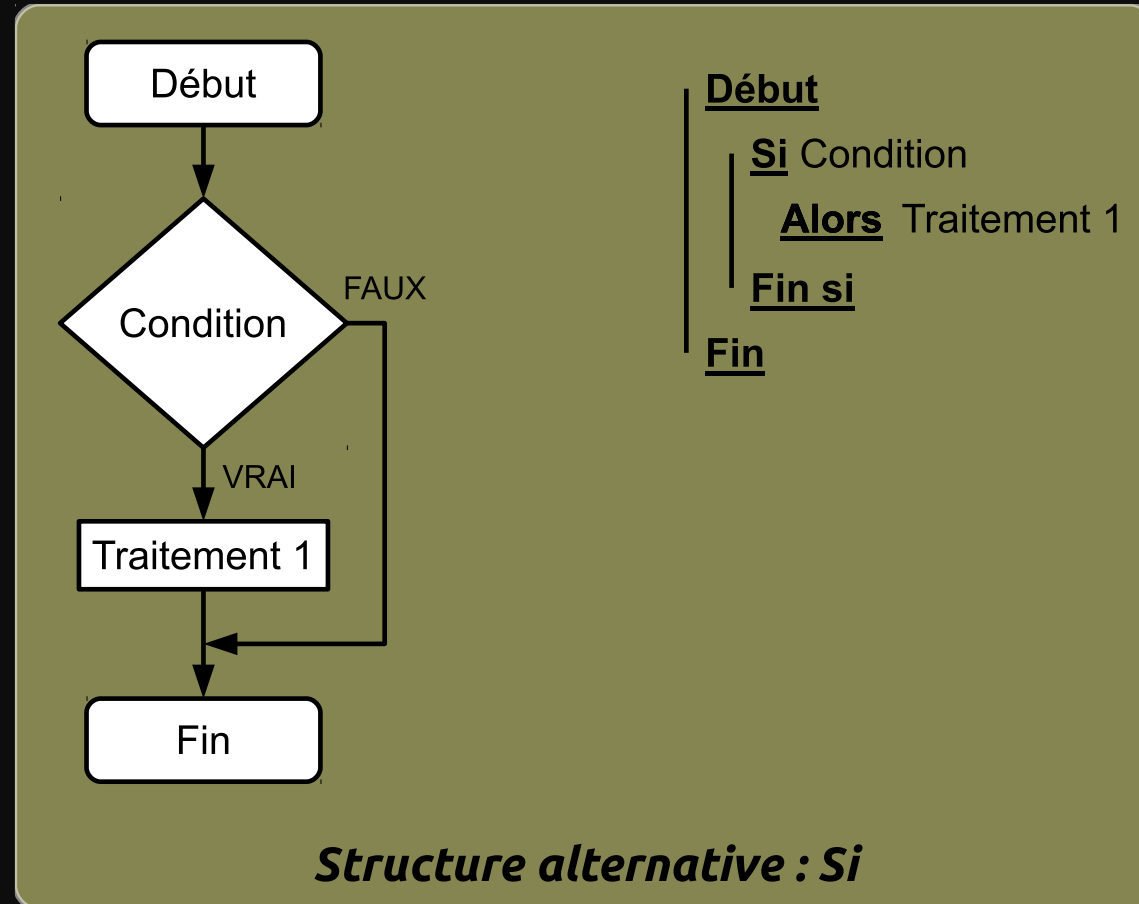
Booléen

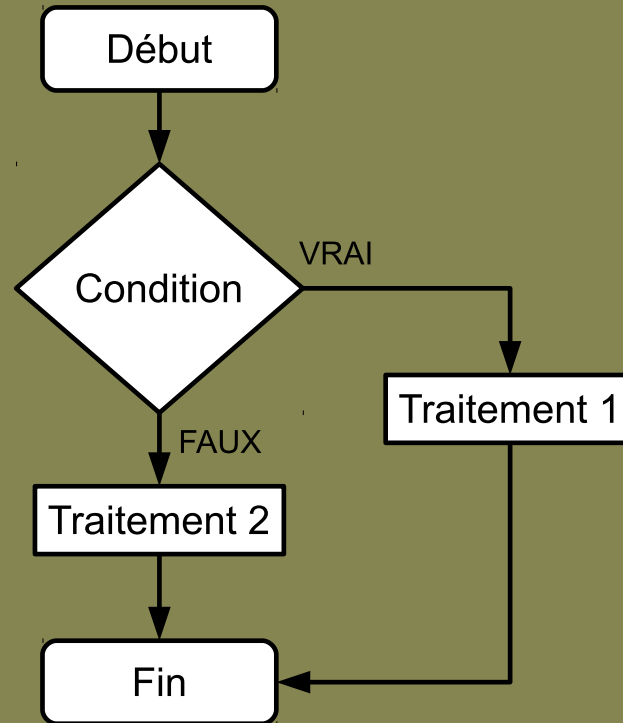
Flottant

Octet

Caractère







Début

Si Condition

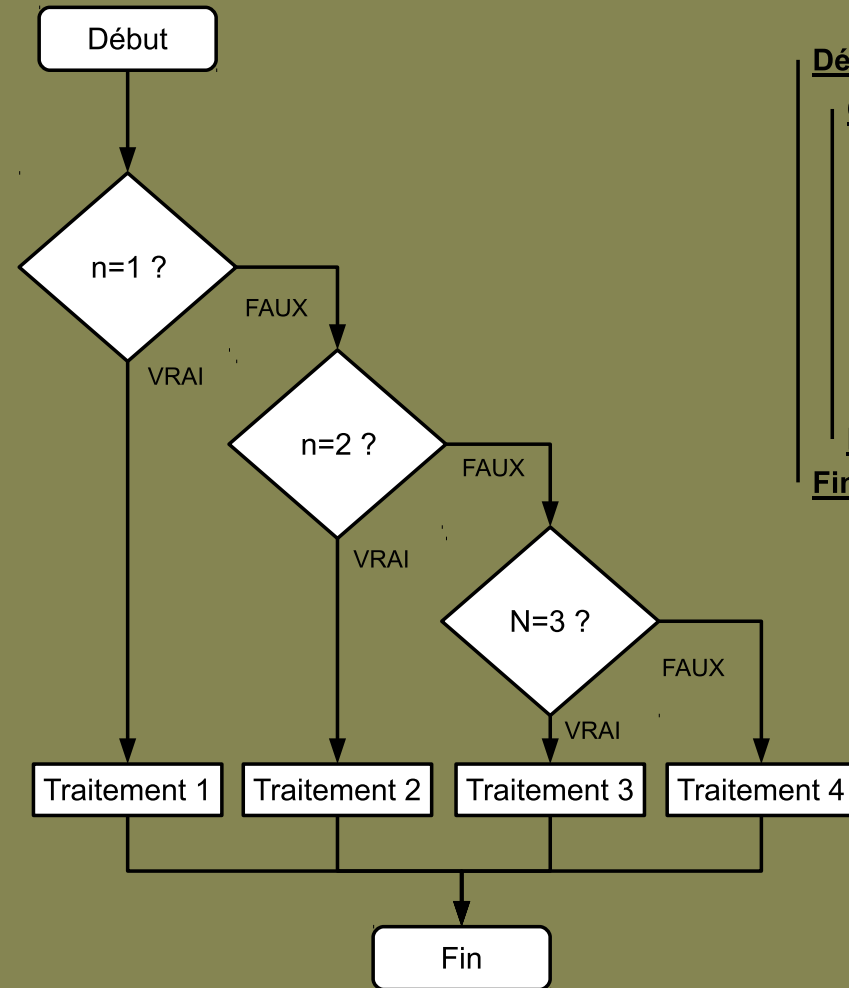
Alors Traitement 1

Sinon Traitement 2

Fin si

Fin

Structure alternative : Si sinon



Début

Choix n

Cas 1 : Traitement 1

Cas 2 : Traitement 2

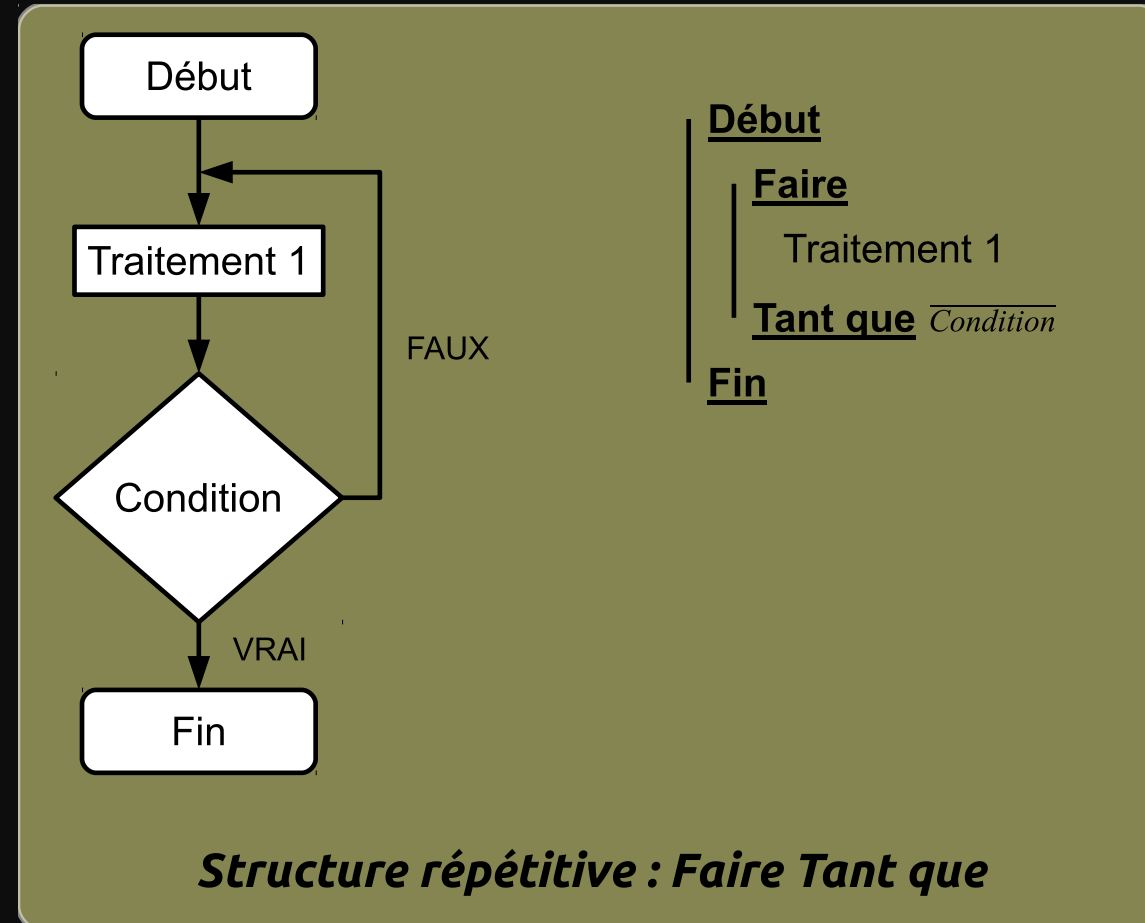
Cas 3 : Traitement 3

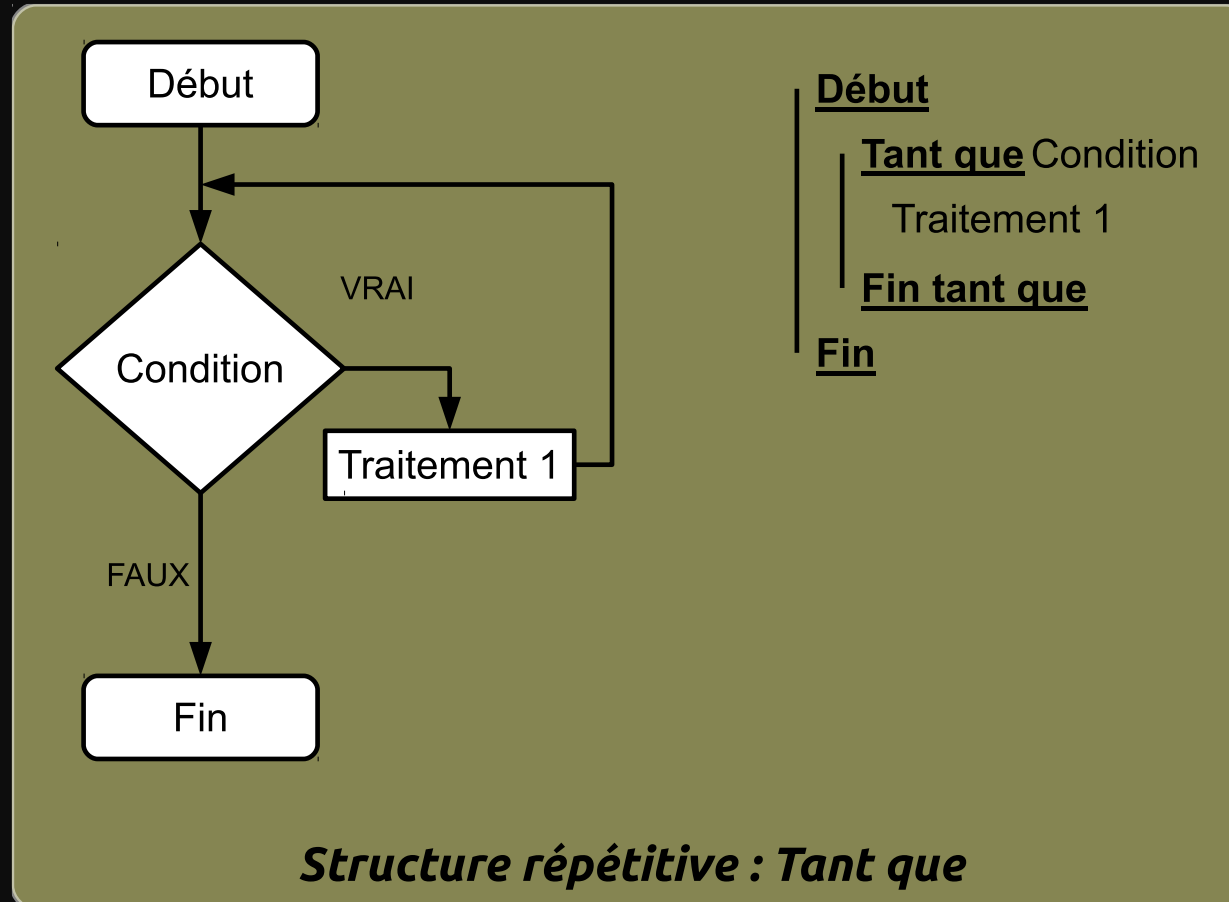
Défaut : Traitement 4

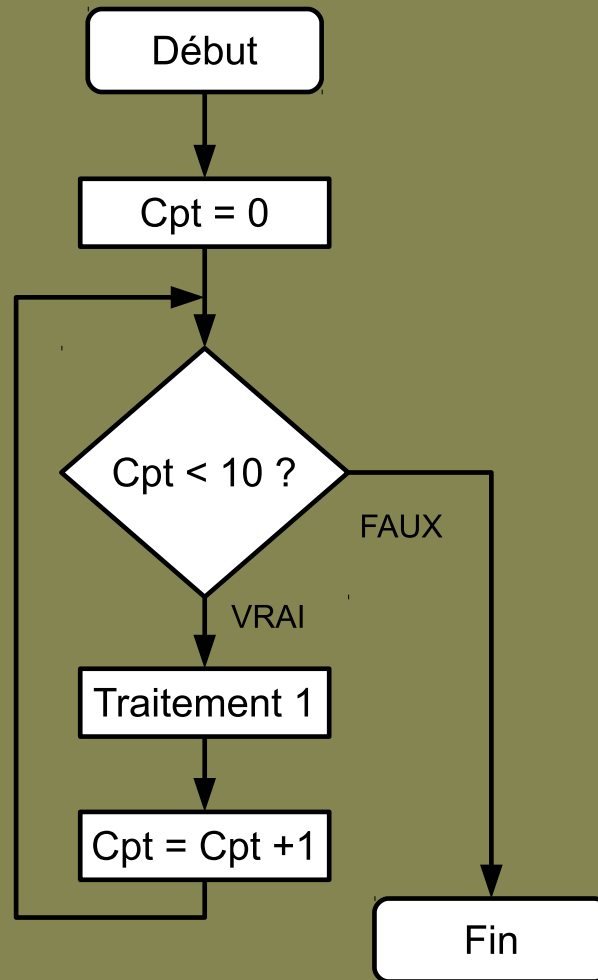
Fin choix

Fin

Structure alternative: Choix multiple







Début

Pour (Cpt=0 , Cpt<10, Cpt=cpt+1)

Traitement 1

Fin pour

Fin

Cpt : Variable de comptage
Cpt = 0 : Indice de début
Cpt < 10 : Indice de fin
Cpt = Cpt+1 : Pas de comptage

Structure répétitive : Pour

