



# Informatique



## Initiation à l'algorithmique pour Visual Basic

**Auteur** : C. Terrier ; <mailto:webmaster@cterrier.com> ; <http://www.cterrier.com>

**Utilisation** : Reproduction libre pour des formateurs dans un cadre pédagogique et non commercial

Ce cours est une initiation à l'algorithmique et à la programmation pour Visual Basic. Il est destiné à des débutants. Il est conçu de la façon suivante :

- **Chapitre 1 - Notions de base** : Ces pages abordent les notions indispensables à connaître pour débiter et progresser dans la création d'algorithme pour Visual Basic
- **Chapitre 2 - Les bases de la programmation** : repose essentiellement sur des exemples d'exercices à lire et à comprendre. Puis à la fin de chaque partie des exercices d'entraînement sont à réaliser en autonomie.

### Sommaire

- 1 - Notions de base
  - 11 – Définition
  - 12 – Exemple d'algorithme
  - 13 – Variables
  - 14 – Calcul et opérations sur les données
  - 15 - Lecture et écriture
- 2 – Les bases de la programmation
  - 21 – Structure séquentielle
  - 22 – Conditions - si
    - 221 Condition simple
    - 222 – conditions imbriquées
    - 223 – Conditions multiples : et, ou
  - 23 Boucles - FOR
    - 231 – Boucles simples
    - 232 – Boucles imbriquées
  - 24 – Tableau - Pour l = 1 to n
  - 25 – Boucles – Tant que et jusqu'à ce que

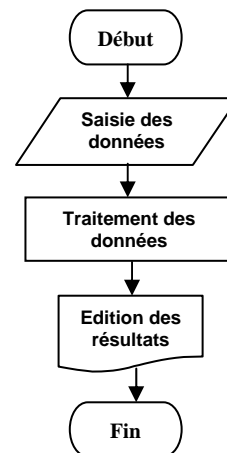
# 1 – Notions de base

Tout travail informatique consiste à **réaliser des traitements** à partir de **données de base** pour obtenir des **données élaborées** (Résultats)

*Exemple : Calculer un salaire mensuel connaissant le taux horaire, le nombre d'heures et le montant de la prime*

L'organisation du travail peut être schématisée de la façon suivante :

De façon plus concrète le travail réalisé sera le suivant :



Etapes	Exemple salaire
<b>Etape 1 : Paramétrage des données initiales qui peuvent être :</b> <ul style="list-style-type: none"> <li>Des constantes prédéfinies (taux de cotisation, taux d'intérêt, plafond de SS etc.)</li> <li>Des données saisies par l'opérateur au clavier</li> </ul>	<b>Paramétrer les données de base</b> <ul style="list-style-type: none"> <li>Nombre d'heures</li> <li>Taux horaire</li> <li>Prime</li> </ul>
<b>Etape 2 : Les données sont traitées pour obtenir les résultats attendus</b> (Ce sont des données élaborées)	<b>Traitement des données</b> <ul style="list-style-type: none"> <li>Salaire horaire = Nombre d'heures * taux horaire</li> <li>Salaire brut = salaire horaire + Prime</li> </ul>
<b>Etape 3 : les données obtenues sont restituées sous forme d'affichages écran ou d'impressions</b>	<b>Edition</b> <ul style="list-style-type: none"> <li>Afficher le salaire horaire</li> <li>Afficher le salaire brut</li> </ul>

## 11 – Définition

**L'algorithme** prépare la **programmation** qui sera réalisés sur l'ordinateur. Il décompose les tâches que le programme doit réaliser de façon logique et chronologique en opérations transcripibles dans un langage informatique qui sera dans le cadre de ce cours **Visual Basic (VBA)**.

## 12 – Exemple d'algorithme

En général un algorithme s'organise de la façon suivante :

Algorithme salaire	Explications
<b>' Paramétrages des variables</b> VAR Heures : Réel 'champ nombre d'heure VAR Txhoraire : Réel VAR Prime : Single VAR Base : Single 'Salaire de base mensuel VAR Brut : Single ' Salaire brut mensuel	Le texte précédé d'un ' correspond à un commentaire  Un champ est une mémoire dans laquelle les données sont saisies. Pour chaque champ il est indiqué : <ul style="list-style-type: none"> <li>la nature du champ =&gt; <b>VAR</b> = Variable</li> <li>le nom du champ =&gt; Heures ou Txhoraire</li> <li>le type du champ =&gt; Réel =&gt; numérique à format long</li> </ul>
<b>Début</b>	Marque le début des traitements sur les variables
<b>'Saisie des données</b> Lire (« saisir le nombre d'heure : » ; Heures) Lire (« saisir le taux horaire : » ; Txhoraire) Lire (« saisir le montant de la prime : » ; Prime)	Tout programme commence par une saisie des données de base qui sera demandé à l'écran pour être saisie au clavier
<b>'Calcul</b> Base = Heures * Txhoraire Brut = Base + Prime	Les calculs utilisent les données qui sont dans les variables complété précédemment
<b>'Edition des résultats</b> Ecrire (« Le salaire de base est de : » ; Base) Ecrire (« Le salaire brut est de : » ; Brut)	Les résultats sont imprimés ou affichés à l'écran
<b>Fin</b>	Marque la fin des traitements

Nous allons à présent aborder plus en détails ces notions.

## 13 – Les variables de données

Les variables sont des mémoires dans lesquelles les données sont placées. Elles sont créées et paramétrées au début de chaque algorithme.

- **Nom**

Il commence obligatoirement par une lettre et ne doit comporter aucun espace. Il est possible de remplacer l'espace par le trait de soulignement de la touche [8] « \_ ». Les caractères suivants sont interdits : -, +, \*, /, ?, !

☺ Les caractères accentués (avec accents : é, è, à, ù) sont fortement déconseillés.

Exemple :     *Capital*           => *Capital*  
               *Prix\_TTC*         => *Prix\_TTC*  
               *12NET*           => *Net12*

- **Types**

Les données peuvent être du texte ou des nombres, elles peuvent être quantitatives (un prix, une quantité, une dimension) ou qualitative (code postal, numéro de sécurité sociale, plaque minéralogique).

Les données quantitatives sont susceptibles d'être utilisées dans des calculs, pour cette raison elles doivent être placées dans des variables numériques alors que les autres informations peuvent être placées dans des variables de type texte ou caractère.

Les types suivants sont proposés en Visual Basic :

	Algorithme	VBA	Limites d'utilisation
<b>Format nombre</b>	Entier	Integer	Entier de -32 768 à 32 767
	Réel	Long	Entier de – 2 milliards à + 2 milliards environ
	Simple	Single	7 chiffres avec virgule flottante
	Double	Double	15 chiffres avec virgule flottante
<b>Booléen</b>	Booléen	Boolean	False ou True
<b>Format texte</b>	Chaîne	String	Maxi 64 000 caractères
<b>Date</b>	Date	Date	
<b>Tout type</b>	Variable	Variant	aucune

☺ Par défaut une variable avec virgule est capable de traiter des entiers.

- **Syntaxe**

Les données saisies ou utilisées peuvent être des **variables** ou des **constantes**.

### Variables :

Les variables reçoivent des contenus qui peuvent changer au cours du traitement. Ils sont précédés du mot : **Var** (algorithme) ou **DIM** (VBA) suivi du type de données.

La syntaxe à utiliser lors de la création du nom de champ est la suivante :

- **Var** (nom du champ) : (type du champ) dans les algorithmes
- **Dim** (nom du champ) **As** (type du champ) en VBA

Exemple :     **Var** Longueur : **Entier**   => création d'un variable longueur de type entier  
               **Var** Prix : **Simple**       => création d'un variable Prix de type simple

### Constantes :

Les constantes reçoivent un contenu qui sera fixe pour tous le traitement. Leur contenu peut être défini dès le début de l'algorithme. Ils sont précédés du mot : **Const**.

Exemple : **Const** TVA : **Simple** = 0,196

Attention la syntaxe change entre l'algorithme et le VBA.

Algorithme	VBA
<b>Var</b> Longueur : <b>Entier</b>	<b>Dim</b> Longueur <b>As Integer</b>
<b>Const</b> TVA : <b>Simple</b> = 0.196	<b>Const</b> TVA <b>As Single</b> = 0.196

Il est possible de paramétrer plusieurs champs à la suite en les séparant par une virgule

Exemple : **Var** Longueur : **Entier**, Largeur : **Entier**, Surface : **Entier** (Attention le type doit être précisé pour chaque variable)

## 14 – Calcul et opération sur les données

Les principaux opérateurs numériques sont :

Opération	opérateurs	Exemples
Addition	+	Longueur + Largeur
Soustraction	-	Prix - Remise
Division	/	Total / Nbre notes
Multiplication	*	Prix * Quantité
Puissance	^	Capital * (1+ taux) ^ durée

Les opérateurs de comparaison sont :

Opération	opérateurs	Exemples
Supérieur	>	Age > 18
Inférieur	<	Age < 18
Inférieur ou égal	<=	Age <= 18
Supérieur ou égal	>=	Age >= 18
Différent	<>	Age <> 18
Affectation d'un résultat	=	Brut = Base + Prime Attention de ne pas confondre l'opérateur de comparaison qui est = Et l'opérateur d'affectation d'un résultat qui est :=

Les opérateurs logiques sont

Opération	opérateurs	Exemples
ET	<b>And</b>	Si (Condition 1) <b>and</b> (condition2) faire
Ou	<b>Or</b>	Si (Condition 1) <b>or</b> (condition2) faire
Non	<b>Not</b>	Si ( <b>not</b> Condition 1) faire

## 15 – Lecture et écriture

Une partie du travail consiste à demander des informations à l'opérateur et à lui restituer

En algorithme on utilise les fonctions Lire et Ecrire en VBA on utilise les valeurs

Nom variable = **InputBox** et **Msgbox**

Algorithme	VBA
<b>Lire</b> (Longueur)	Longueur = <b>InputBox</b> (« Saisir la longueur : »)
<b>Ecrire</b> (Surface)	Surface = <b>MsgBox</b> (« La surface est de : »)

Il est possible d'ajouter un texte dans les fonctions lire et écrire

- Lire (« **Saisir votre nom** : » ; Nom)
- Ecrire (« **Le résultat est de** : » ; Résultat)

Il est possible de regrouper plusieurs information sur une même ligne à l'aide du symbole de concaténation &

- Ecrire (« **Le chiffre d'affaires de la société** : » ; Client & « est de : » CA)