



Dominique Maniez

FORMATION À...

VBA

Toutes versions
de Office 2000
à Office 2007

**Visual Basic pour Applications
pour Word, Excel,
PowerPoint, Access et Outlook**

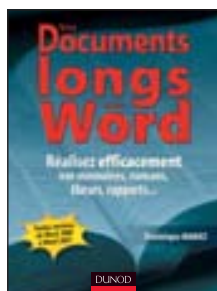
DUNOD

FORMATION À VBA

Consultez nos catalogues sur le Web



www.dunod.com



Vos documents longs avec Word
Réalisez efficacement vos mémoires,
romans, thèses, rapports...
Dominique Maniez
192 p.
Dunod, 2007.

Programmation et algorithmique en VBA pour Excel

Anne Brygoo, Maryse Pelletier,
Michèle Soria et Séverine Dubuisson
240 p.
Dunod, 2007.



FORMATION À VBA

Dominique Maniez

Développeur, journaliste et universitaire

DUNOD

Toutes les marques citées dans cet ouvrage sont des marques déposées par leurs propriétaires respectifs.

:

Le pictogramme qui figure ci-contre mérite une explication. Son objet est d'alerter le lecteur sur la menace que représente pour l'avenir de l'écrit, particulièrement dans le domaine de l'édition technique et universitaire, le développement massif du photocopillage.

Le Code de la propriété intellectuelle du 1^{er} juillet 1992 interdit en effet expressément la photocopie à usage collectif sans autorisation des ayants droit. Or, cette pratique s'est généralisée dans les établissements

d'enseignement supérieur, provoquant une baisse brutale des achats de livres et de revues, au point que la possibilité même pour

les auteurs de créer des œuvres nouvelles et de les faire éditer correctement est aujourd'hui menacée. Nous rappelons donc que toute reproduction, partielle ou totale, de la présente publication est interdite sans autorisation de l'auteur, de son éditeur ou du Centre français d'exploitation du

droit de copie (CFC, 20, rue des Grands-Augustins, 75006 Paris).



© Dunod, Paris, 2007
ISBN 978-2-10-050872-3

Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5, 2° et 3° a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

Table des matières

Avant-propos	XIII
------------------------	------

Partie 1 Apprendre à programmer

Chapitre 1 – Qu’est-ce que programmer ?	3
Plusieurs niveaux de programmation	5
Les langages de programmation	6
La syntaxe	7
Les phases de conception d’un programme	8
<i>La phase d’étude préalable</i>	<i>8</i>
<i>La phase d’analyse</i>	<i>9</i>
<i>La phase d’encodage</i>	<i>10</i>
<i>La phase de test</i>	<i>10</i>
<i>La phase de production</i>	<i>10</i>
Chapitre 2 – Enregistrer une macro	13
L’enregistreur de macros	14
<i>Quand devez-vous enregistrer une macro ?</i>	<i>14</i>
<i>Enregistrement de votre première macro</i>	<i>15</i>
Exécuter une macro	17
Où sont stockées les macros	18
Comment assigner un raccourci clavier à une macro ?	21
Comment associer une icône à une macro ?	23
Conseils pour l’enregistrement des macros	26

Le choix du nom des macros	27
<i>Les limitations de l'enregistreur de macros</i>	29
Enregistrement d'une macro avec Excel	32
Chapitre 3 – Modifier le code des macros	37
Voir le code de la macro	38
Modifier le code de la macro	47

Partie 2

Le langage VBA

Chapitre 4 – Syntaxe de VBA	55
Historique de VBA	55
Différences entre Visual Basic et VBA	56
Syntaxe de VBA	57
Variables	58
Constantes	63
Opérateurs	65
Mots clés	68
<i>Instructions</i>	70
Chapitre 5 – Variables et tableaux	73
Types de données	73
<i>Les dates</i>	76
<i>Les caractères</i>	79
<i>Les nombres</i>	80
<i>Le type de données Variant</i>	82
<i>Les erreurs de type</i>	82
<i>Les expressions</i>	83
Visibilité des variables	85
Tableaux	88

Chapitre 6 – Tests conditionnels	91
Les tests conditionnels	91
<i>If Then Else</i>	92
<i>Traiter plus de deux choix</i>	94
<i>Opérateur logique dans une condition</i>	96
Imbriquer des tests conditionnels	97
Select Case	99
Chapitre 7 – Boucles	103
For Next	103
<i>Sortir de la boucle</i>	109
While Wend	111
Do Loop	114
<i>Expression logique</i>	116
Null	117
Empty	118
Gare aux boucles infinies	119
Différences entre While et Until	121
Chapitre 8 – Procédures et fonctions	125
Procédures et fonctions	126
Syntaxe d'une fonction	127
<i>MsgBox en détail</i>	132
<i>Fonctions de Visual Basic par catégorie</i>	145
<i>Écrire ses propres fonctions</i>	161
<i>Les paramètres facultatifs</i>	166

Partie 3

Modèles d'objets

Chapitre 9 – Objets	171
Définition d'un objet	172

Objets dans Office	173
Un objet en situation	174
Écrire des fonctions pour manipuler des objets	185
L'explorateur d'objets	186
Autres modèles d'objets	188
Chapitre 10 – Programmer Word	191
Objet Application	192
Objet Document	196
Objet Range	199
Objet Selection	203
Mise en pratique	207
Pilotage d'une application Office à partir d'une autre application	211
Chapitre 11 – Programmer Excel	217
Objet Application	217
Objet Workbook	223
Objet Worksheet	225
Objet Range	227
Mise en pratique	233
Chapitre 12 – Programmer Access	237
Collections d'Access	238
Objets d'Access	239
Objet DoCmd	241
Objet Form	245
Mise en pratique	250
<i>Apparition et remplissage d'une liste par programmation</i>	<i>251</i>
<i>Remplir un champ automatiquement</i>	<i>255</i>

Chapitre 13 – ADO	257
Installation d'ADO	258
Objets d'ADO	259
Objet Connection	261
Objet Recordset	263
Mise en pratique	265
Exemples Access	265
Exemple Word	269
Exemple Excel	272
Exemples d'utilisation d'un fichier MDB sans Access	274
Chapitre 14 – Programmer Outlook	283
Modèle d'objets	283
Objet MailItem	285
Objet MAPIFolder	287
Accès à un sous-dossier à partir de la Boîte de réception	289
Mise en pratique	291
Envoyer un message à partir d'une BD	292
Analyser tous les messages entrants	293
Exporter les messages dans une BD	294
Exporter les contacts dans une BD	296
Chapitre 15 – Programmer PowerPoint	299
Objet Application	301
Collection Presentations	303
Collection Slides	310
Collection Shapes	315
Numéros et énumérations	319
Mise en pratique	324

Chapitre 16 – XML	329
Introduction	329
De SGML à XML en passant par HTML	330
<i>Histoire d'une norme</i>	331
<i>Description de SGML</i>	332
<i>Objectif de SGML</i>	333
<i>Une DTD particulière : HTML</i>	334
<i>Description de XML</i>	337
<i>Objectifs de XML</i>	337
XML en action	339
Le nouveau format des fichiers Office 2007	343

Partie 4

Programmation VBA avancée

Chapitre 17 – Créer des formulaires	349
Exemple de UserForm pas à pas	350
Mise en pratique	364
<i>Création du UserForm</i>	368
Chapitre 18 – Gérer des fichiers texte	375
Objet FileSystemObject	376
Objet TextStream	377
Mise en pratique	381
<i>Conversion de fichiers au format Vcard</i>	381
<i>Créateur de fichiers batch</i>	384
Chapitre 19 – Programmer les API	387
Concept d'API	388
Declare	389
Appel de fonction API	390

Mise en pratique	391
<i>Lecture d'un fichier WAV</i>	391
<i>Lecture d'un fichier Midi</i>	392
<i>Récupérer des informations</i> <i>sur la configuration vidéo</i>	393
<i>Macro globale de recherche-remplacement</i>	395
Chapitre 20 – Déboguer un programme	401
Erreurs de programmation	401
<i>Erreurs de syntaxe</i>	402
<i>Erreurs d'exécution</i>	404
<i>Erreurs de logique</i>	410
Débogage	410
Débogueur	413
<i>Lancement du débogueur</i>	413
<i>Fonctionnement du débogueur</i>	414
<i>Visualisation des variables dans le débogueur</i>	420
Gestion des erreurs	421
Chapitre 21 – Aller plus loin	425
Organiser les macros	425
Prendre de bonnes habitudes	427
Se documenter	430
Index	437

Avant-propos

J'utilise le traitement de texte Word depuis sa deuxième version sous DOS, ce qui signifie qu'entre les versions en mode texte et les versions sous Windows, j'ai connu une bonne dizaine de versions différentes. J'ai toujours été fasciné par la puissance de cet outil avec lequel je travaille quasi quotidiennement. J'ai aussi très vite découvert que la puissance intrinsèque de Word pouvait être décuplée par l'utilisation d'un mini-langage, baptisé à l'époque, macro-commandes. Il n'était pas rare de constater que quelques macro-commandes d'une quinzaine de lignes pouvaient économiser des heures de travail pénible. L'écriture de ces petits bouts de programme nécessitait parfois du temps mais on avait toujours la sensation, une fois la macro-commande finalisée, d'avoir travaillé plus intelligemment que si l'on avait dû réaliser la tâche à accomplir manuellement. En fait, la plupart des utilisateurs de Word sont tellement absorbés par leur travail d'écriture qu'ils négligent totalement de lire la documentation du produit et passent à côté des fonctions les plus puissantes de ce traitement de texte. Le même constat s'applique bien évidemment à Excel, ainsi qu'aux autres applications de la suite Office.

En rédigeant ce livre, j'ai voulu démontrer à tous les utilisateurs d'Office qu'ils se privent inutilement de la richesse fonctionnelle de leur traitement de texte, de leur tableur ou de leur base de données en ignorant la programmation. En vous apprenant à programmer Word, Excel, Access, Outlook et PowerPoint, je souhaite premièrement vous montrer que cette activité n'est pas réservée aux professionnels de l'informatique et, deuxièmement, vous faire gagner du temps dans l'exécution des tâches répétitives et fastidieuses.

À QUI S'ADRESSE CE LIVRE ?

Cet ouvrage est un livre d'initiation et il ne nécessite donc aucune connaissance préalable en programmation ; il vise par conséquent un public de débutants. Il s'adresse d'abord aux utilisateurs de la suite Office qui souhaitent aborder l'apprentissage de la programmation afin d'améliorer leur productivité. Les personnes utilisant Office et possédant déjà une expérience de programmeur peuvent également profiter de ce livre en faisant l'économie de la lecture des chapitres consacrés aux rudiments de la programmation. Cet ouvrage n'est pas un ouvrage de référence en ce sens où il ne prétend absolument pas à l'exhaustivité ; de nombreuses informations sont sciemment passées sous silence afin de clarifier le propos et de ne pas semer la confusion dans l'esprit du lecteur par un apport trop important de connaissances nouvelles (à titre d'information, l'ouvrage intitulé Microsoft Word 2000 Language Reference comporte près de 1 500 pages...).

La démarche pédagogique mise en œuvre dans ce livre est similaire à la méthode de programmation qui procède par raffinements successifs ; cette méthode reprend en fait un principe cartésien qui stipule qu'il faut commencer « par les objets les plus simples et les plus aisés à connaître, pour monter peu à peu comme par degrés jusqu'à la connaissance des plus composés ». La dernière partie de cet ouvrage proposera, à ceux qui le souhaitent, des pistes pour qu'ils puissent approfondir les sujets abordés dans ces pages ou bien explorer d'autres horizons plus complexes de la programmation sous Office.

POURQUOI APPRENDRE À PROGRAMMER OFFICE ?

Il peut paraître étonnant de se poser une telle question dans un ouvrage d'initiation à la programmation Office. En effet, les gens qui souhaitent s'initier à une discipline particulière connaissent en général très bien leurs motivations. Pourtant, il existe tellement de malentendus au sujet de la programmation que j'ai souhaité clarifier certains points et également tordre le cou à des idées reçues qui sont fort répandues. Je pense également que de nombreuses personnes

n'osent pas s'aventurer dans l'apprentissage de la programmation par peur de ne pas y arriver. Dans cet avant-propos, je souhaite déculpabiliser tous ceux qui en ont envie, mais qui n'osent pas sauter le pas.

Qu'est-ce qui peut bien pousser un utilisateur de micro-ordinateur à se lancer dans l'apprentissage de la programmation ? Je vais ici faire un tour d'horizon de toutes les raisons qui peuvent influencer quelqu'un à vouloir s'initier à VBA qui est le langage de programmation d'Office. Quand j'aurai terminé l'inventaire de tous ces motifs, vous constaterez qu'ils sont tellement nombreux qu'il n'y a vraiment aucune raison valable de s'en priver.

La première raison est d'ordre intellectuel. Apprendre à programmer, c'est devenir acteur du processus informatique. Quand on programme, on est moins passif devant sa machine et on acquiert une meilleure connaissance du fonctionnement matériel et logiciel de l'ordinateur. En même temps, on acquiert certains types de raisonnements logiques qui peuvent servir dans d'autres domaines que celui de la programmation.

La deuxième raison est culturelle. Quand vous voulez comprendre la culture d'un pays étranger, il est nécessaire de connaître les rudiments de la langue parlée par les autochtones. Il en va de même pour l'informatique : vous devez apprendre un langage de programmation pour mieux comprendre la culture informatique qui, au fil du temps, a pénétré tous les secteurs de la société.

La troisième raison est sociale. Aujourd'hui, l'informatisation de la société est poussée à l'extrême et c'est finalement un enjeu social que de comprendre comment les programmes fonctionnent. Nous sommes un certain nombre à penser que la chose informatique ne doit pas être l'apanage des informaticiens. En apprenant un langage de programmation, vous empiétez sur le territoire des informaticiens et vous vous attaquez ainsi à leur toute puissance, ce qui est démocratiquement sain car il est anormal qu'une caste d'individus ait autant de pouvoirs sans rendre de comptes à personne.

La quatrième raison est productiviste. La programmation, même à un niveau peu élevé, va vous permettre de gagner un temps précieux, surtout si vous accomplissez des tâches répétitives. En effet, l'automatisation des tâches va augmenter votre productivité, parfois dans des

proportions que vous n'imaginez même pas. Outre le gain de temps, vous allez également vous affranchir des tâches pénibles et pouvoir ainsi vous consacrer à des tâches plus nobles. Au final, vous constaterez que l'amélioration est non seulement quantitative, mais également qualitative.

La dernière raison est qu'en programmant vous allez pouvoir bénéficier d'un logiciel sur mesure car vous allez créer tout ce qui vous manque. Les possibilités de paramétrage d'Office sont déjà importantes, mais en programmant, vous allez contrôler exactement les traitements de votre système d'information. Apprendre à programmer ouvre des horizons quasiment infinis et il est bien difficile d'être exhaustif si on veut inventorier toutes les applications pratiques. Nous nous contenterons ici de quelques exemples. En maîtrisant les rudiments de la programmation, vous allez déjà pouvoir inventer des commandes et des fonctions qui n'existent pas dans le logiciel (par exemple des fonctions d'Excel qui vous manquent).

Vous allez pouvoir également contrôler la validité des informations qui sont saisies dans Word, Excel ou Access. Dans tous ces logiciels, il est extrêmement facile de saisir des données mais dès que l'on veut exercer un contrôle minimal sur les informations qui sont saisies, il faut avoir recours à la programmation. Et si on réfléchit bien, on s'aperçoit qu'il est inutile de traiter des données par de savants calculs si on n'a pas pris la précaution de s'assurer de la validité de ces informations.

De la même manière, si vous développez des modèles qui doivent être utilisés par d'autres, la programmation vous aidera à définir des écrans d'aide spécifiques ou bien des formulaires de saisie personnalisés qui faciliteront la tâche de ceux qui doivent entrer les informations.

Enfin, et c'est ce qui est sans doute le plus simple, vous automatiserez tous les traitements répétitifs. C'est d'ailleurs souvent dans l'automatisation des tâches banales que la programmation se révèle d'une efficacité maximale, et l'apprenti programmeur est toujours étonné du gain de productivité fantastique que peuvent lui procurer quelques lignes de code.

Ainsi, après avoir lu cet ouvrage :

- vous aurez une bonne idée de ce qu'est la programmation ;
- vous maîtriserez les concepts de base de la programmation ;
- vous saurez écrire de petits programmes sous Office ;
- vous aurez tordu le cou à des mythes encore vivaces ;
- vous pourrez vous lancer dans l'apprentissage d'un langage de programmation plus puissant.

Importance des exemples de code

Il est impossible de concevoir un ouvrage traitant de la programmation Office sans de nombreux exemples de code car, si l'on apprend à programmer en programmant, on étudie également la programmation en examinant le code de programmes écrits par d'autres. Imprimer le code de tous les exemples au sein de cet ouvrage ne serait guère raisonnable car cela prendrait une place considérable ; il est d'autre part prouvé que la recopie d'un listing imprimé engendre de nombreuses erreurs de retranscriptions. C'est pour cette raison que ne sont imprimés dans ce livre que de courts exemples ou bien des extraits de programmes plus longs. Cependant, il faudra absolument que vous vous procuriez la totalité des exemples de code de cet ouvrage qui sont disponibles sur Internet sur le site des éditions Dunod à l'adresse suivante :

www.dunod.com

ou bien sur mon site personnel (dans la rubrique Code des ouvrages) à l'adresse suivante :

www.cosi.fr

UN OUVRAGE VRAIMENT CONÇU POUR LES DÉBUTANTS

C'est peut-être parce que je n'arrivais pas à trouver les livres que j'avais envie de lire que je me suis mis à en écrire. Cela ne veut pas dire que mes livres sont meilleurs que les autres, mais tout simplement qu'ils correspondent mieux à ce que je recherche.

Quand j'ai commencé à apprendre à programmer, il y a de cela une vingtaine d'années, j'ai dévoré des dizaines de livres sur le sujet. Même aujourd'hui, quand j'apprends un nouveau langage de program-

mation, je lis des ouvrages de programmation. Après toutes ces années passées à lire cette littérature technique sur la programmation, je suis arrivé à la conclusion qu'il n'existait pas véritablement d'ouvrage conçu pour les débutants qui n'y connaissent rien du tout. Les livres de programmation sont avant tout conçus pour les informaticiens ; cette démarche peut se comprendre dans la mesure où la plupart des informaticiens programment, mais comme notre credo est que tout le monde peut programmer et que la programmation ne doit surtout pas être réservée aux informaticiens, il existe un véritable problème pour les lecteurs qui ne sont pas informaticiens, mais qui souhaitent néanmoins s'initier à la programmation. Ce livre a donc pour but de s'adresser aux gens qui n'y connaissent vraiment rien et qui veulent découvrir les joies (et les peines) de la programmation avec Office. Cet objectif implique que la pédagogie mise en œuvre dans cet ouvrage prenne véritablement en compte le caractère néophyte du lecteur. Je ne prendrai qu'un seul exemple qui illustre bien cette différence de traitement pédagogique ; dans les livres de programmation, il est nécessaire d'apprendre la syntaxe (c'est-à-dire la grammaire) du langage de programmation étudié. En général, tous les livres commencent par décrire la syntaxe formelle, puis prennent des exemples. Nous sommes persuadés que cette méthode ne fonctionne pas avec des débutants qui ne sont pas habitués au formalisme de la description de la syntaxe du langage. Nous pensons au contraire qu'il faut commencer par les exemples et éventuellement passer au formalisme après avoir étudié de nombreux exemples.

POURQUOI APPRENDRE LA PROGRAMMATION DE CINQ LOGICIELS EN MÊME TEMPS ?

Dans les premières versions d'Office, chaque logiciel de la suite avait son propre langage et les langages de programmation étaient donc incompatibles entre eux ; ainsi, par exemple, Word Basic n'était pas compatible avec Access Basic. Avec l'avènement d'Office 2000, Microsoft a réalisé un effort considérable d'harmonisation et désormais, VBA (Visual Basic pour Applications) est l'unique langage de programmation de la suite. Ce qui signifie que quand j'apprends à programmer Word, je sais programmer à la fois Excel, Access, Out-

look et PowerPoint. L'unicité de ce langage est un progrès énorme et c'est pour cette raison qu'il serait dommage de se limiter à l'apprentissage de la programmation d'un seul logiciel quand il est si facile de passer d'un logiciel à l'autre.

L'apprentissage de VBA représente donc un très bon investissement car, si vous n'utilisez, par exemple, que Word et Excel, vous pourrez très facilement apprendre la programmation Access étant donné que le langage est rigoureusement le même. Le fait qu'Office propose un même langage pour toutes ses applications est réellement un avantage déterminant et nous pensons qu'il va inciter plus d'un utilisateur à se lancer dans l'aventure de l'apprentissage de la programmation VBA.

COMMENT APPRENDRE A PROGRAMMER OFFICE ?

Au risque de rappeler une évidence, pour apprendre à programmer Office, il faut déjà apprendre Office. Cette vérité première mérite d'être répétée tant on a vu d'utilisateurs se lancer dans l'apprentissage de la programmation sans maîtriser les fonctionnalités élémentaires de Word (comme par exemple, les styles, les modèles ou bien encore les tableaux), d'Excel (écriture d'une formule, adresse relative ou absolue, etc.) ou d'Access (création de tables, de requêtes ou de formulaires). Si vous pensez que vos connaissances d'Office sont imparfaites, il faudra donc les approfondir et nous vous conseillons pour ce faire la lecture des ouvrages consacrés aux applications de la suite Office, la collection « Au quotidien » chez Microsoft Press constituant un bon point de départ.

Une fois que ces connaissances sont acquises, il faut apprendre le langage de programmation VBA et le modèle d'objets des applications Office. Nous emploierons ici souvent l'analogie avec l'apprentissage des langues vivantes et l'ambition de ce livre est donc de vous enseigner la syntaxe (le langage VBA) et le vocabulaire (le modèle d'objets) de chacun des logiciels de la suite afin que vous puissiez écrire vous-même rapidement des programmes.

Il existe cependant une difficulté importante quand on veut apprendre une langue étrangère : par où commencer ? La tâche semble immense (elle l'est réellement) et la logique voudrait qu'avant de s'exprimer on commence par maîtriser la grammaire et le lexique. Mais cette approche pédagogique est bien peu efficace et chacun d'entre nous se rend bien compte que l'on apprend une langue en la pratiquant, la théorie ne pouvant venir que dans un deuxième temps. Nous allons donc apprendre à programmer en programmant et nous étudierons la théorie seulement quand nous en aurons réellement besoin.

QUELLE VERSION D'OFFICE FAUT-IL UTILISER AVEC CE LIVRE ?

Cet ouvrage couvre le langage de programmation des versions d'Office allant de la version 2000 à la version 2007. Outre sa dénomination commerciale, chaque version d'Office comporte un numéro de version interne qu'il est préférable de connaître. Le tableau suivant établit la correspondance entre ces deux dénominations :

Appellation commerciale	Numéro de version
Office 2000	9
Office XP	10
Office 2003	11
Office 2007	12

Office 2007 introduit quelques différences notables entre les précédentes versions d'Office du point de vue de la programmation. Ces différences sont signalées dans cet ouvrage par des encadrés labellisés Nouveauté Office 2007.

PARTIE 1

Apprendre à programmer

1

Qu'est-ce que programmer ?

Nombreux sont ceux qui pensent que c'est la connaissance de la programmation qui fait la différence entre l'informaticien et l'utilisateur averti. Ainsi, de nombreux utilisateurs chevronnés du traitement de texte craignent de se lancer dans l'apprentissage de la programmation en pensant qu'il s'agit là d'un monde qui leur est inaccessible. L'ambition de ce livre est de démontrer qu'ils ont tort et que la programmation, abordée en douceur et avec pédagogie, n'est pas l'apanage des professionnels de l'informatique ; en effet, l'utilisateur lambda, s'il maîtrise les bases de la logique, peut apprendre aisément à programmer. Cette entreprise est à la portée de tous et cet ouvrage prétend démythifier la programmation, en montrant tout d'abord que cette discipline de l'informatique repose sur des techniques que chacun utilise dans la vie courante. Cela signifie que, comme Monsieur Jourdain faisait de la prose sans le savoir, vous avez déjà programmé, même si vous l'ignorez.

Nous définirons tout d'abord la programmation comme l'art d'écrire des programmes et nous dirons qu'un programme est une suite d'instructions. Le Grand Robert donne une définition plus complète que je vous livre ci-dessous :

« Ensemble ordonné des opérations nécessaires et suffisantes pour obtenir un résultat ; dispositif permettant à un mécanisme d'effectuer ces opérations. »