

## Les langages de programmation

Marcel Labelle

Numéro 50, mai 1983

L'ordinateur à l'école

URI : <https://id.erudit.org/iderudit/55407ac>

[Aller au sommaire du numéro](#)

### Éditeur(s)

Les Publications Québec français

### ISSN

0316-2052 (imprimé)

1923-5119 (numérique)

[Découvrir la revue](#)

### Citer cet article

Labelle, M. (1983). Les langages de programmation. *Québec français*, (50), 64–69.

Exemple: LOGO, PASCAL, APL, BASIC, SMALLTALK, ...

#### Matériel

équipement informatique.

Exemples: ordinateur, écran, clavier, imprimante, ...

#### Mémoire morte

(ROM read only memory)

la zone de l'ordinateur où sont stockées en permanence les instructions vitales qui ne s'effacent pas quand la machine est fermée.

#### Mémoire vive

(RAM random access memory)

la zone de l'ordinateur dans laquelle les données, les instructions et les résultats sont stockés temporairement et qui s'effacent lorsque l'alimentation est coupée.

#### Microprocesseur

unité de traitement contenue sur une plaquette (ou puce) de silicium et capable d'effectuer diverses tâches à l'intérieur de l'ordinateur.

#### Ordinateur

un appareil doté d'un microprocesseur qui perçoit des données, les traite suivant un programme mémorisé et communique les résultats de ses calculs.

#### Périphérique

tout élément d'un ordinateur distinct de l'unité centrale de traitement, susceptible de lui être branché et d'être commandé par elle.

Exemples: imprimante, écran, table traçante, lecteur de disquette, ...

#### Programme

ensemble d'instructions expliquant à l'ordinateur ce qu'il doit faire.

#### Puce de silicium

minuscule plaquette de silicium pur traité chimiquement et portant des circuits électroniques miniaturisés. Elle remplit à elle seule le rôle de plusieurs centaines de composantes électroniques.

#### Traitement de texte

Ensemble des opérations où le système informatique permet de stocker des textes et de les faire apparaître à l'écran afin de les modifier et de les réviser avant la frappe définitive.

#### Terminal

unité périphérique telle qu'un écran ou une imprimante servant à l'introduction des données et l'affichage des résultats.

#### Unité centrale de traitement (UCT)

centre de contrôle de l'ordinateur qui répartit les tâches. C'est en quelque sorte le « cerveau » sans la mémoire.

#### Videotex

système informatique permettant aux abonnés d'avoir accès à des informations contenues dans des bases de données. Ces informations sont affichées sur un téléviseur.

Exemples: TÉLIDON, ANTIOPE, PRESTEL.

# Les langages de programmation

marcel labelle



Pour traiter des informations et obtenir des résultats bien précis, on peut commander un ordinateur avec des mots tels que PRINT, ÉCRIS, AVANCE, BEGIN, GOTO, IF... THEN... GOSUB, etc. Ces mots ainsi choisis à partir de l'anglais ou du français constituent le vocabulaire d'un langage de programmation.

Les langages humains sont complexes et imprécis. Si l'on communique à l'ordinateur: « Belle machine, vos beaux yeux me font mourir d'amour », il affichera laconiquement SYNTAX ERROR. Il ne pourrait répondre.

Ces mots ne font pas partie de son vocabulaire et il ne peut interpréter comme vous un si beau compliment. Toute langue naturelle (français, anglais, japonais...) englobe un trop grand nombre de concepts pour être utilisable par un ordinateur. Pouvez-vous imaginer l'ensemble de règles d'interprétation qu'il faudrait donner à la machine pour qu'elle comprenne les notions de beauté, de nuances ou de sentiments? Il faut lui donner des ordres précis et univoques: AVANCE de 100 pas, tourne à DROITE de 60 degrés, ADDITIONNE

ces nombres, AFFICHE sur écran le texte qui suit: ...

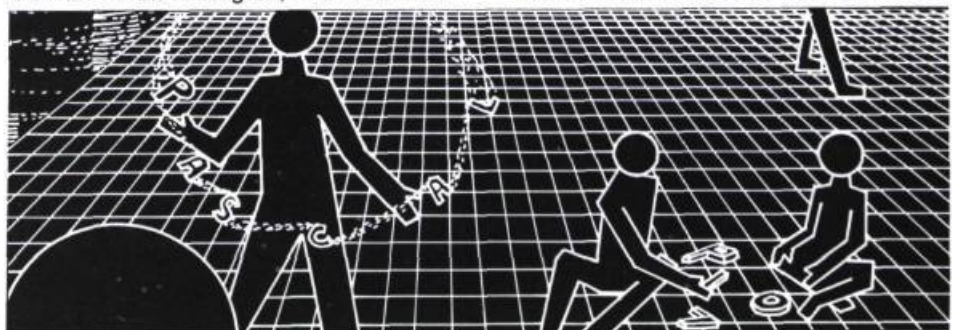
Ce serait extraordinaire si les ordinateurs pouvaient comprendre et utiliser les langages humains. Mais l'interprétation de tels langages par une machine dépasse pour le moment, et probablement pour bien longtemps encore, les capacités d'un système informatique.

Lorsqu'on parle de langage pour les ordinateurs, on pense le plus souvent à BASIC. Mais d'autres langages ont été développés, notamment LOGO, PASCAL, FORTH, PILOT ou APL, qui devraient beaucoup faire parler d'eux à l'avenir. On a même créé des langages français LSE (langage symbolique d'enseignement) et le BASICOIS, une version française de BASIC. De plus, ces langages sont maintenant accessibles aux micro-ordinateurs. Le choix est vaste. Nous ne pouvons pas tous les passer en revue mais nous vous présenterons BASIC, LOGO, PASCAL et PILOT.

Tout langage de programmation est défini

- par un **vocabulaire**, c'est-à-dire un ensemble de mots orthographiés correctement

en BASIC: PRINT, INPUT, IF... THEN,





en LOGO: AVANCE, REPETE, SAUF  
PREMIER, ÉDITE, ÉCRIS...

en PASCAL: VAR, BEGIN, IF...  
THEN..., ELSE

en PILOT: T:, A:, M:, J:, ...

- par une **syntaxe**, c'est-à-dire un ensemble de règles de grammaire,

exemple: en BASIC IF 4 < A < 10  
THEN PRINT A

est un énoncé qui ne respecte pas la  
règle de formation; il faudra écrire  
IF A > 4 OR A < 10 THEN PRINT A

- et par une **sémantique**, c'est-à-dire un ensemble de règles d'interprétation

exemple: en LOGO, la procédure pour  
faire un triangle peut s'écrire:

POUR TRIANGLE

— AVANCE 100

— DROITE 120

— AVANCE 100

— DROITE 120

— AVANCE 100

FIN

et on obtient un triangle équilatéral  
ayant trois angles de 60 degrés. Si on  
écrit DROITE 60 au lieu de DROITE  
120, nous n'obtiendrons pas le résultat  
voulu.

### Programme

Pour communiquer avec un ordina-  
teur, on doit lui fournir une suite finie  
d'instructions. Ceci constitue un pro-  
gramme.

Un programme doit être rédigé pour  
préciser à l'ordinateur ce qu'il faut faire.  
Un programme est l'expression, dans  
un langage de programmation détermi-  
né, d'une procédure, interprétable  
par le système informatique. L'ordina-  
teur ne résout pas le problème; c'est le  
travail du programmeur. Pour lui, écrire  
un programme consiste à concevoir et à  
exprimer des procédés de résolution de  
problèmes dans un nombre fini d'étapes,  
puis à les traduire dans un langage  
compréhensible par la machine. Le lan-  
gage dans lequel les programmes sont  
écrits comprend un certain nombre  
d'opérations simples. L'ordinateur qui  
comprend ce langage est capable d'ef-  
fectuer les opérations correspondantes.

Nous pouvons maintenant expliquer  
comment l'ordinateur traite les données.  
Pour ce, le programmeur rédige une  
solution en trois grandes étapes. La  
première étape est celle de la prépa-  
ration du traitement. Le programmeur  
indique les données nécessaires à la  
résolution du problème. La deuxième  
étape est celle du traitement. Le pro-  
grammeur formule pas à pas la réso-  
lution du problème après décomposi-  
tion de celui-ci. La troisième étape est  
celle de l'affichage des résultats. L'or-  
dinateur affiche sur écran les résultats  
obtenus par l'exécution de l'étape pré-  
cédente. En résumé:

données --> traitement --> résul-  
tats.

### Pseudo-programme

Puisque l'ordinateur est une machine  
sans imagination et sans intelligence, la  
procédure doit être présentée d'une  
manière claire et sans ambiguïté. Il n'est  
pas nécessaire d'être mathématicien  
pour écrire un programme d'ordinateur,  
mais il faut être habile à expliquer les  
choses clairement. Il s'agit en fait de  
communiquer sa pensée avec le mini-  
mum de mots et le maximum de clarté  
et de cohérence.

Je vous explique avec un exemple:  
«Aimez-vous les crêpes?... Aimez-vous  
la bière?... Si vous répondez «oui» à  
ces deux questions, vous aimerez sans  
doute ma recette de crêpes à la bière.»  
Imaginez, pour un instant, que vous  
disposez d'un robot répondant aux  
commandes suivantes: VERSEZ, CAS-  
SEZ, REMUEZ, AJOUTEZ, SI... ALORS,  
SINON, PARFUMEZ. Votre robot peut  
traiter les données comme la farine, les  
œufs, le lait, l'huile, le sucre, le sel et  
même la bière. Il vous reste maintenant  
la tâche de communiquer cette recette  
à votre robot.

L'analogie que je vous propose res-  
pecte les règles semblables à celles  
d'un langage de programmation. J'ex-  
plique cette recette à l'aide d'un pseudo-  
langage qui est une étape intermédiaire  
souvent utilisée avant la codification  
dans un langage de programmation



**Le CEC est fier d'annoncer qu'il  
est diffuseur exclusif, en milieu  
scolaire, des  
logiciels**

**LOGIDISQUE**

entre autres:

**BASIC FRANÇAIS** L.-P. Hébert, B. Piché (Tous âges)

Un logiciel qui permet de rédiger des programmes en français,  
interprète les programmes écrits en anglais et en fournit une liste  
en **BASIC FRANÇAIS**; toutes les communications ordinateur-  
utilisateur se font en français. Un outil essentiel de program-  
mation française! 2 disques 69,95 \$

**S.E.D. 3.3** L.-P. Hébert, P. Olivier (Tous âges)

Système d'exploitation de disques bilingue qui permet de lire  
et d'utiliser indifféremment les instructions en français ou en an-  
glais. 1 disque 39,95 \$

**VERBES ET PROVERBES** L. Leblanc (Âges: 8-12 ans)

Un didacticiel qui permet l'apprentissage des conjugaisons et  
des proverbes français. 1 disque 39,95 \$

**pour APPLE II plus, APPLE II e,  
DOS 3.3, 48 K**

**CEC**



déterminé. Chaque instruction s'exprime avec un verbe d'action et s'écrit en respectant des règles de présentation et d'écriture. Par exemple, tout énoncé se termine par le point virgule sauf le dernier: FIN, qui peut être suivi d'un point. Le lecteur avisé verra peut-être une certaine ressemblance avec un programme écrit en PASCAL.

#### PROGRAMME CRÊPE

données

##### INGRÉDIENTS:

160 grammes de farine,  
3 œufs,  
½ litre de lait,  
1 c. s. d'huile,  
200 grammes de sucre,  
2 c. s. de bière,  
un peu de sel;

traitement

#### DÉBUT

VERSEZ la farine dans un plat;  
CASSEZ 3 œufs;  
REMUEZ les œufs et la farine;  
AJOUTEZ une pincée de sel;  
VERSEZ l'huile;  
VERSEZ le lait petit à petit;  
REMUEZ le tout;  
Si la pâte est épaisse, ALORS  
AJOUTEZ du lait;  
REMUEZ le tout;

SINON  
Si la pâte est trop liquide  
ALORS  
AJOUTEZ de la farine;  
REMUEZ le tout;

Si vous désirez une pâte légère  
ALORS

AJOUTEZ un peu de bière;  
REMUEZ le tout;

SINON  
PARFUMEZ à votre choix  
LAISSEZ reposer une heure

résultat

LA PÂTE EST PRÊTE;

FIN.

L'analogie s'arrête ici. C'était pour vous faire comprendre la notion d'algorithme ou de programme. Il ne faut pas s'imaginer que vous pouvez communiquer cette recette à votre robot culinaire, même s'il est doté d'un magnifique microprocesseur.

### Quelques langages

#### BASIC

(Beginners All Purpose Symbolic Instruction Code)

BASIC fut inventé en 1963 au Collège Dartmouth au New Hampshire par les professeurs Kemeny et Kurtz dans le but de rejoindre les élèves de disciplines autres que l'informatique. Conçu à une période où les langages de programmation étaient encore fort proches de la structure des machines, il a connu un développement étonnant. Outil d'enseignement, il est devenu un outil de programmation sur tous les micro-ordinateurs. C'est un langage interactif facile à apprendre, très populaire et on le retrouve sur tous les micro-ordinateurs. En mode dialogué, il permet d'établir une sorte de conversation entre l'ordinateur et l'utilisateur car au moment de soumettre le programme à l'ordinateur, il peut répondre aux questions posées, inscrire les données requises et il peut même vérifier rapidement la logique de son raisonnement, car il obtient immédiatement des résultats. Si ces derniers sont faux, il peut rapidement modifier une ou des instructions et soumettre à nouveau son programme à l'ordinateur. Cela est vrai pour tous les langages dits interactifs comme BASIC ou LOGO.

Un programme BASIC est constitué d'une série d'énoncés numérotés qui sont interprétés et exécutés dans l'ordre, un à la fois.

Mon but n'est pas de vous donner une initiation au BASIC mais plutôt d'illustrer la façon de présenter un programme à l'ordinateur avec ce langage.

BASIC a ses qualités et ses défauts. Plusieurs le critiquent à cause de son manque de structure. Pour comprendre cette notion, c'est un peu comme si vous écriviez un texte sans jamais faire de paragraphe. Un programme complexe sans documentation appropriée

Exemple:

#### Instructions du programme

#### Notes explicatives

	10 PRINT «Quel est ton prénom?»	← Question qui sera affichée à l'écran.
donnée →	20 INPUT N\$	← Cette instruction permettra l'enregistrement du prénom en mémoire.
	30 PRINT N\$; « , écris trois nombres »	← Les commandes sont en anglais mais on peut faire afficher du texte en français.
données →	40 INPUT A, B, C « ,	← Cette instruction permettra l'enregistrement de 3 nombres en mémoire.
traitement →	50 S = A + B + C	← L'ordinateur additionnera les 3 nombres inscrits, retiendra le résultat dans un emplacement de mémoire représenté par S (pour somme).
	60 M = S/3	← Pour obtenir la moyenne M, l'ordinateur divisera la somme S par 3.
résultat →	70 PRINT «La moyenne de ces 3 nombres est: »;M	← Instruction pour faire afficher cette phrase et le résultat obtenu M.
	80 END	← Fin du programme.

#### RUN

← Pour commander l'exécution de ce programme par l'ordinateur.

#### Dialogue affiché à l'écran:

(ordinateur)	Quel est ton prénom?	← et ensuite RETURN.
(usager)	Eusèbe	
(ordinateur)	Eusèbe, écris trois nombres.	← L'utilisateur inscrit 3 nombres en les séparant par des virgules
(usager)	25, 30, 35.	suivi de RETURN.
(ordinateur)	La moyenne de ces 3 nombres est: 30.	← Résultat affiché.

devient vite illisible et incompréhensible. D'autres lui reprochent d'utiliser trop librement l'instruction GOTO (aller à) pour modifier la séquence d'instructions et d'inculquer de mauvaises méthodes d'analyse et de programmation. Il a été conçu à une époque où les principes d'une programmation correcte n'avaient

pas encore été mis en valeur. Malgré ces inconvénients, ce langage a connu un succès étonnant, si bien qu'on trouve un dialecte de BASIC sur à peu près tous les micro-ordinateurs. Pour utiliser un autre langage, il faut alors payer un supplément, et de plus avoir l'appareil approprié.



## LOGO

(construire une procédure)

LOGO a été conçu par une équipe du (MIT), sous la direction de Seymour Papert, dans le but de transformer l'éducation grâce à la technologie de l'ordinateur. Ce nom est dérivé du mot grec « LOGOS » et est utilisé au MIT depuis 1971 pour désigner un projet à la convergence des recherches en intelligence artificielle et en sciences de l'éducation. C'est à la fois un langage, une théorie d'apprentissage et un ensemble d'équipements informatiques constituant un environnement riche pour l'enfant qui veut résoudre les problèmes qu'il se pose.

Pour les tenants de l'approche LOGO, le travail de l'élève ne consiste pas à trouver la « bonne réponse ». Il est plus important de construire une procédure explicite pour faire trouver la solution d'un problème qu'il a choisi plutôt que de trouver soi-même cette solution ; ainsi l'élève analyse ses propres procédures et ses démarches. Pour cela, il part d'hypothèses, les soumet à l'ordinateur, compare les résultats que celui-ci donne avec ce qu'il attendait.

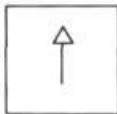
LOGO est donc construit à des fins pédagogiques. C'est un langage simple et il utilise les possibilités de graphiques. Une flèche placée sur l'écran symbolisant la « tortue » peut avancer, reculer, tourner et former des figures.

Expliquons d'abord comment l'élève peut commander la tortue.

1. La tortue est placée au centre de l'écran



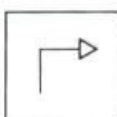
2. S'il écrit AVANCE 100, la tortue avance de « 100 » « pas ».



3. S'il écrit DROITE 90, la tortue se retourne à 90 degrés.



4. S'il écrit AVANCE 100, la tortue avance encore de 100 pas.



Donc en répétant 4 fois AVANCE 100 DROITE 90 l'ordinateur affichera un carré.

Pour montrer un nouveau mot à l'ordinateur, on résumerait les instructions précédentes de la façon suivante :

Pour CARRÉ

RÉPÈTE 4 [AVANCE 100 DROITE 90]

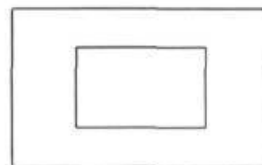
OU

RÉPÈTE 4 [AV 100 DR 90]

FIN

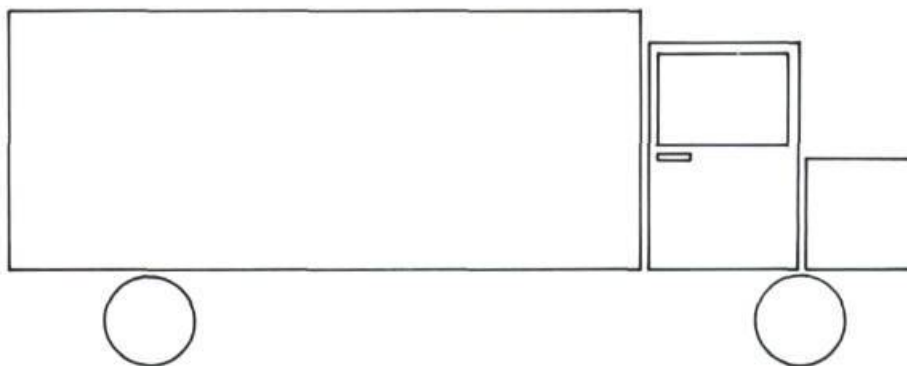
— 4, 100, 90 sont les données  
— RÉPÈTE, AVANCE, DROITE représentent le traitement

Résultat :



Le langage LOGO permet le développement de divers concepts. La proposition et la modification d'une solution, par exemple, permettent à l'élève de soumettre à l'observateur une solution à un problème qu'il se pose. Il peut nommer les objets, décomposer un problème en sous-problèmes et construire une solution à l'aide des objets créés.

Supposons qu'un élève de quatrième année du primaire désire faire tracer par l'ordinateur le camion que voici :



Problème

Sous-problèmes

Sous-sous-problèmes

CAMION

BOÎTE  
CABINE  
MOTEUR  
ROUES

FENÊTRE  
POIGNÉE

AVANT  
ARRIÈRE

1. Procédure  
POUR CAMION  
BOÎTE  
CABINE  
FENÊTRE  
POIGNÉE  
MOTEUR  
ROUEAVANT  
ROUEARRIÈRE

FIN

2. Procédure  
POUR ROUEAVANT  
RÉPÈTE 360 [AVANCE 1 DROITE 1]  
FIN

Ces procédures peuvent être considérées comme un ensemble de « briques » à partir desquelles il est possible de construire la procédure CAMION. Il apprend à structurer sa pensée et à s'exprimer clairement.

En conclusion, LOGO permet à l'élève d'agir en proposant des solutions, de réagir en les modifiant et de conserver les résultats de son expérience. Avec le langage LOGO, l'élève exploite d'abord le graphisme mais il peut faire autre chose. Il peut effectuer des calculs et traiter du texte comme les autres langages de programmation.



# PPMF LE PPMF-UQAM:

## UN PROGRAMME

### POUR VOUS!

L'UQAM offre aux enseignants du primaire un programme PPMF centré sur la pratique quotidienne et réalisé selon un modèle pédagogique d'autodidaxie assistée.

En s'impliquant dans un modèle d'autodidaxie assistée l'enseignant peut entre autres:

- développer son autonomie;
- apprendre à son rythme;
- élaborer et réaliser des projets d'intervention adaptés à sa classe.

Pour plus d'informations:  
(514) 282-3635



Université du Québec à Montréal

## PILOT

(Programmed Inquiry Learning Or Teaching)

PILOT est un langage d'auteur, c'est-à-dire un langage spécialement conçu pour que l'enseignant puisse devenir auteur de séquences programmées. Il a été développé à l'Université de Cali-

fornie de San Francisco. Langage interactif qu'on retrouve sur certains micro-ordinateurs (Apple, Atari et autres), il permet à une personne sans trop d'expérience en programmation de créer des dialogues pour l'enseignement.

Une situation comme celle qui suit se présente très souvent durant une leçon sur ordinateur.

S'il te plaît, écris ton nom et appuie sur la touche RETURN

■ MARCEL

- l'élève inscrit son nom et appuie sur RETURN dès qu'il a terminé.

Maintenant, Marcel, quel est ton âge ?

■ 20

- suivi de RETURN

Quel est le triple de ton âge ?

■ 60

- suivi de RETURN

Écris un homonyme de MER

■ MAIRE

- suivi de RETURN

Un exemple de programme rédigé avec PILOT.

## Instructions du programme

partie de  
programme

D: N\$ (50)

T: S'il te plaît écris ton nom et appuie sur la touche RETURN

A: \$N\$

K: NOM: \$N\$

T: Maintenant, \$N\$, quel est ton âge ?

A: #A

JE: ERREUR

K: âge: #A

C: R = 3 \* A

procédure

\* PROBLÈME

T: Quel est le triple de ton âge ?

A: #B

JE: ERREUR

T1: (B=R)

procédure

\* HOMONYME

T: Écris un homonyme de MER

A:

M: MÈRE! MAIRE

procédure

\* ERREUR

En utilisant PILOT, l'enseignant peut par exemple présenter un texte à l'élève,

## Commentaires

D: pour «DIMENSION»; l'ordinateur réservera l'espace nécessaire pour 50 caractères.

T: pour «TEXT», c'est-à-dire la question qui s'affichera à l'écran.

A: pour «ACCEPT»; l'ordinateur acceptera la réponse de l'élève.

K: pour «KEEP»; l'ordinateur conservera le nom dans un fichier;

\$N\$ (variable) sera remplacé par le nom écrit précédemment; l'ordinateur acceptera une réponse numérique.

JE: pour «JUMP TO ERROR»; l'ordinateur passera à la procédure ERREUR (#A: variable); l'ordinateur conservera l'âge dans un fichier.

(R: variable); C: pour «COMPUTE» et pour le symbole de la multiplication; l'ordinateur calculera le triple de l'âge.

\*: pour identifier une procédure.

question qui s'affichera à l'écran.

(#B: variable); l'ordinateur acceptera une réponse numérique.

Si la réponse n'est pas un nombre, l'ordinateur passera à la procédure ERREUR.

T1: pour «TRUE», si la réponse est acceptable au premier essai.

= question affichée à l'écran.

A: pour accepter un mot comme réponse.

M: pour «MATCH», c'est-à-dire comparer la réponse à la réponse acceptable donnée par l'élève et! pour «OU».

lui laisser suffisamment de temps pour l'étudier et ensuite l'interroger sur la





compréhension du texte à l'aide de questions simples ou à choix multiples. L'élève écrit sa réponse. L'ordinateur vérifie sa justesse. Dans le cas contraire, il lui donne soit une autre chance et de l'information supplémentaire, soit la bonne réponse suivie d'un commentaire et la prescription d'un correctif. Ces dialogues doivent être souples et variés. L'auteur peut même les illustrer avec des graphiques et les agréments avec de la musique électronique. PILOT possède des facilités graphiques et sonores simples à manipuler.

## PASCAL

PASCAL est un langage de programmation dérivé de l'ALGOL. Or l'ALGOL était un langage de programmation très populaire en Europe mais difficile à apprendre. En 1971, Niklaus Wirth de Zurich invente PASCAL comme outil didactique pour enseigner l'ALGOL et démontrer les principaux éléments d'un langage structuré. PASCAL est un langage approprié pour définir les structures de données afin de résoudre des problèmes. Le langage a été nommé ainsi en l'honneur de Blaise Pascal, mathématicien français qui a inventé la première machine à calculer.

PASCAL était surtout utilisé sur des gros ordinateurs, mais comme ce langage gagne de plus en plus de popularité, on le retrouve aussi sur la plupart

des micro-ordinateurs si on accepte de payer un supplément.

Un programme écrit en PASCAL est plus facile à lire et à comprendre qu'un programme en BASIC. Il exprime clairement et d'une manière concise la solution d'un problème. On dit que c'est un excellent langage pour exprimer des algorithmes et qu'il est particulièrement approprié pour enseigner de bonnes méthodes de programmation.

Il y a bien quelques critiques. Certaines sont d'ordre technique et débordent largement le cadre de notre propos. D'autres sont importantes pour celui qui désire s'initier à la programmation. Pour un débutant, ce n'est pas un langage facile à maîtriser.

## Conclusion

À côté des besoins des spécialistes, il apparaît nécessaire, d'une part, d'introduire une large initiation à l'informatique afin d'intégrer l'usage de cet outil à son enseignement sans pour autant être informaticien; d'autre part, cette formation doit s'appuyer sur une sensibilisation à l'informatique destinée à contribuer à la culture générale de tous les citoyens.

Or cette sensibilisation ne peut se faire sans parler, entre autres choses, de langages de programmation. Mais voilà, lequel choisir? Tout dépend des objectifs pédagogiques visés. Si vous

désirez faire programmer l'élève (oui, c'est possible dans un cours de français) et développer des habiletés intellectuelles, LOGO semble tout désigné.

Par contre, si vous voulez préparer vos propres didacticiels rapidement et efficacement, un langage d'auteur comme PILOT est plus indiqué. On pourrait ainsi trouver des arguments justifiant l'utilisation d'autres langages. Ce que vous devez retenir, c'est qu'un langage de programmation est un outil de communication qui inculque des méthodes d'analyse qui lui sont propres.

Il n'est pas facile de faire des recommandations quant aux choix des langages de programmation sans déclencher une avalanche de discussions et de protestations. C'est la Tour de Babel. Tous clament que leur choix est le meilleur et qu'il convient mieux à l'apprentissage et à leurs besoins. Il faut plutôt conserver une attitude critique sans pour autant tomber dans le piège des guerres de langages. De nouveaux naissent régulièrement et remplaceront les langages comme BASIC.

Je vous invite à surveiller l'apparition d'un langage, dont on parle depuis près de dix ans, qui devrait transformer la conception classique de la programmation. SMALLTALK a été développé au centre XEROX à Palo Alto en Californie par Alan C. Kay et son équipe. SMALLTALK, c'est un nom à retenir.

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

**DYNAMIQUE...**

## REGARD NEUF SUR L'APPRENTISSAGE

### AUTEURS :

Raymonde Picard  
Germaine Pouliot  
Monique Pouliot

### DYNAMIQUE I

- 4 livres de lecture
- Cahiers d'activités
- Guide pédagogique
- Matériel complémentaire

**DYNAMIQUE I** respecte la démarche proposée par le *MÉQ.*

Le matériel propose :

- des amorces aux thèmes et aux textes ;
- des intentions de lecture ;
- des pistes d'objectivation ;
- des activités de décontextualisation et de réinvestissement ;
- les 4 types de discours ;
- les 4 entrées en lecture.

**DYNAMIQUE I** favorise l'esprit de *découverte* et de *créativité* chez l'enfant.



Éditions Études Vivantes  
6700, chemin Côte de Liesse  
Saint-Laurent (Québec) H4T 1E3  
Tél.: (514) 341-6690

**DISPONIBLE POUR  
LA RENTRÉE**