



# LISP

UNE INTRODUCTION  
A LA PROGRAMMATION

H. WERTZ

BIBLIOTHEQUE DU CERIST



MASSON

MANUELS INFORMATIQUES MASSON

# LISP

## Une introduction à la programmation

**Harald WERTZ**

Professeur à l'Université Paris VIII - Vincennes  
Chargé de recherche au CNRS



BIBLIOTHEQUE DU CERIST

MASSON

Paris New York Barcelone Milan  
Mexico Sao Paulo  
1985

## AVANT-PROPOS

LISP est un des plus anciens langages de programmation : ses premières implémentations ont vu le jour à la fin des années 60, quelques années seulement après FORTRAN.

LISP a été conçu par John McCarthy pour le traitement d'expressions symboliques. Dès son origine, il a été utilisé pour écrire des programmes de calcul symbolique différentiel et intégral, de théorie des circuits électriques, de logique mathématique et de la programmation de jeux.

LISP est également un des langages de programmation les plus répandus : il existe pratiquement sur tous les ordinateurs, de toute taille et de toute origine.

LISP est un des langages de programmation les plus vivants. Le nombre étonnant de versions différentes, telles que MACLISP, INTERLISP, Common-LISP, VLISP, LE\_LISP, etc, en témoigne. De plus, signalons l'apparition de machines utilisant une architecture spécialement conçue pour l'implémentation de LISP, par exemple les ordinateurs de Symbolics ou de la LISP-Machine Company, ou l'ordinateur Maia, actuellement en développement à la CGE.

En LISP, de même que dans les langages-machine, la représentation des données et la représentation des programmes sont identiques : ainsi un programme LISP peut construire d'autres programmes LISP ou se modifier au fur et à mesure de son exécution. Surtout, cette identité de représentation permet d'écrire LISP en LISP même.

LISP est un langage interactif avec un environnement de programmation intégré. Ceci implique non seulement que tous les programmes et toutes les données sont accessibles, modifiables et analysables en LISP même, sans sortir du langage, mais également que la construction des programmes s'accélère considérablement : le cycle 'édition de texte → compilation → édition de liens → chargement → exécution' des langages compilés classiques se réduit au cycle 'lecture → évaluation'.

Ces vertus, ainsi que la simplicité de la syntaxe de LISP, la possibilité de

---

---

programmer immédiatement un problème sans passer par des stades de déclarations de variables ou de types, et finalement le fait que LISP soit fondé sur la récursivité, font de LISP un excellent langage d'apprentissage et d'enseignement de la programmation.

L'apprentissage de LISP est le sujet de ce livre d'introduction à la programmation. Il ne présuppose aucune connaissance de la part du lecteur. A travers des exemples commentés il introduit le langage LISP, ses structures de données, ses structures de contrôle et sa programmation.

Comme nous l'avons indiqué ci-dessus, LISP se présente dans des versions diverses : bien qu'il y ait régulièrement des efforts pour trouver un standard de LISP, le développement de ce magnifique langage semble difficile à arrêter et le programmeur LISP travaille dans des *dialectes* spécifiques. Les dialectes utilisés dans ce livre sont d'une part VLISP, le dialecte LISP développé à l'Université Paris VIII (Vincennes) par Patrick Greussay, et, d'autre part, LE\_LISP, le dialecte issu de VLISP et de Common-LISP développé par Jérôme Chailloux à l'INRIA. Ces deux dialectes sont les versions de LISP les plus répandues en France. Lorsque des différences existent entre ces deux dialectes, elles sont indiquées dans des notes en bas de page.

Chaque chapitre se compose de deux parties : la première consiste en une présentation des nouveaux concepts abordés dans le chapitre, la deuxième est une suite d'exercices dont toutes les solutions sont exposées à la fin du livre. Ces exercices font partie intégrante de chaque chapitre. Souvent les chapitres présupposent une compréhension des exercices des chapitres précédents.

Voici le plan du livre :

- Les trois premiers chapitres introduisent les structures de données standard de LISP, atomes et listes, ainsi que les fonctions de base permettant leur accès et leur construction. Sans une compréhension de ces structures et de ces fonctions nous ne pouvons écrire aucun programme.
  - Au chapitre 4 nous introduisons la possibilité pour l'utilisateur de se définir ses propres fonctions. C'est la base de toute écriture de programme LISP. Nous y abordons également les questions de liaisons des variables.
  - Le chapitre 5 présente les prédicats les plus courants ainsi que la fonction de sélection **IF**.
  - Le chapitre 6 introduit à travers quatre exemples la notion de récursivité, la forme de répétition la plus standard de LISP.
  - Le chapitre 7 présente les fonctions arithmétiques et les bases de la programmation numérique en LISP.
-

- Aux chapitres 8 et 9 nous reprenons la notion d'atome et introduisons l'utilisation des P-listes pour attacher à un atome de multiples valeurs. Nous y traitons également une application particulière des P-listes : les fonctions mémorisant leurs activités.
- Le chapitre 10 traite toutes les questions concernant les fonctions d'impression.
- Les chapitres 11 et 12 exposent la différence entre les fonctions de type `EXPR` et de type `FEXPR`. Nous y examinons également les fonctions **EVAL** et **APPLY**.
- Au chapitre 13 nous reprenons des questions concernant les fonctions d'entrée/sortie ainsi que le rôle des divers caractères spéciaux de LISP.
- Au chapitre 14 nous construisons une petite fonction de filtrage.
- Les chapitres 15 et 16 examinent en détail la structure des listes. Nous y verrons également les fonctions d'affectation et un troisième type de fonction : les macro-fonctions.
- Le chapitre 17 présente diverses formes de répétitions itératives.
- Au chapitre 18 nous reprenons l'exemple de la fonction de filtrage. Nous y construisons une fonction de filtrage très puissante, intégrant toutes les notions abordées précédemment.
- Finalement, le chapitre 19 donne les solutions à tous les exercices que vous avez rencontrés dans ce livre.

Nous ne couvrons pas l'ensemble des fonctions disponibles en LISP : seules les plus importantes sont exposées. Pourtant le sous-ensemble de fonctions traitées est largement suffisant pour construire n'importe quel programme sur n'importe quel système.

Nous n'abordons pas non plus des questions d'implémentation de LISP, de sa compilation et - surtout - de son application en Intelligence Artificielle. Ces parties seront traitées dans un deuxième volume. Néanmoins, dans la bibliographie, à la fin de ce livre, nous citons également des livres concernant ces autres aspects de LISP.

Le contenu de ce livre correspond à un cours d'introduction à la programmation en langages évolués que l'auteur assure depuis un certain nombre d'années à l'Université Paris VIII (Vincennes). Je tiens à remercier les étudiants, qui, par leurs remarques critiques, par leurs questions et par leur enthousiasme, ont largement influencé l'écriture de ce livre. C'est à eux tous que ce livre est dédié.

Ce livre a été édité par l'auteur à l'IRCAM sur ordinateur DEC VAX-780 sous

---

système UNIX, grâce à la bienveillance de Pierre Boulez et David Wessel. Sa réalisation n'aurait pas été possible sans le soutien de l'équipe technique de l'IRCAM, notamment de Michèle Dell-Prane et de Patrick Sinz.

Evidemment, ce livre a bénéficié des suggestions, critiques et remarques d'un grand nombre de personnes. Qu'il me soit permis ici de remercier particulièrement Jean-Pierre Briot, Annette Cattenat, Frédéric Chauveau, Pierre Cointe, Gérard Dahan, Yves Devillers, Jacques Ferber, Patrick Greussay, Daniel Goossens, Eric Halle, Christian Jullien, Jean Mehat, Gérard Nowak, Gérard Paul, Yves-Daniel Pérolat, Jean-François Perrot, Christian Riesner, Nicole Rocland, Bernard-Paul Serpette, Jacqueline Signorini, Patrick Sinz, Ellen Ann Sparer, Roger Tanguy.

## TABLE DES MATIERES

1. PROLEGOMENES.....	1
1.1 EXERCICES.....	4
2. LES FONCTIONS DE BASE: QUOTE, CAR, CDR, CONS.....	5
2.1 EXERCICES.....	9
3. UN AUTRE POINT DE VUE SUR LES LISTES.....	10
3.1 EXERCICES.....	13
4. LA DEFINITION DE FONCTIONS.....	15
4.1 EXERCICES.....	19
5. DES PREDICATS ET DE LA SELECTION.....	21
5.1 QUELQUES PREDICATS.....	22
5.1.1 exercices.....	24
5.2 LA FONCTION IF.....	25
5.2.1 quelques exercices.....	27
6. LA REPETITION.....	29
6.1 LA FONCTION REVERSE.....	29
6.2 LA FONCTION APPEND.....	32
6.3 LA FONCTION EQUAL.....	35
6.4 LA FONCTION DELETE.....	38
6.5 EXERCICES.....	40
7. L'ARITHMETIQUE.....	42
7.1 LES FONCTIONS ARITHMETIQUES DE BASE.....	42
7.2 LES PREDICATS NUMERIQUES.....	44
7.3 LA PROGRAMMATION NUMERIQUE.....	46
7.4 EXERCICES.....	52
8. LES P-LISTES.....	54
8.1 LES SYMBOLES.....	54
8.2 L'ACCES AUX P-LISTES.....	55
8.3 EXERCICES.....	58
9. LES MEMO-FONCTIONS.....	60
9.1 EXERCICES.....	63
10. LES ENTREES / SORTIES (PREMIERE PARTIE).....	65
10.1 LA FONCTION GENERALE D'IMPRESSION : PRINT.....	65
10.2 LES CHAINES DE CARACTERES.....	70
10.3 LES FONCTIONS PRIN1, TERPRI ET PRINCH.....	72
10.4 L'ACCES AUX NOMS DES ATOMES.....	77

10.5	EXERCICES .....	81
11.	LES DIVERS TYPES DE FONCTIONS UTILISATEURS: DE et DF .....	84
11.1	LA FONCTION DE .....	84
11.2	LA FONCTION DF .....	87
11.3	EXERCICES .....	88
12.	LES FONCTIONS EVAL ET APPLY .....	90
12.1	LA FONCTION EVAL .....	91
12.2	LA FONCTION APPLY .....	93
12.3	EXERCICES .....	96
13.	LES ENTREES / SORTIES (DEUXIEME PARTIE) .....	97
13.1	LA FONCTION GENERALE DE LECTURE : READ .....	97
13.2	LES AUTRES FONCTIONS D'ENTREE / SORTIE .....	100
13.3	LES DIVERS TYPES DE CARACTERES .....	109
	13.3.1 détermination du type d'un caractère .....	109
	13.3.2 les macros-caractères .....	111
13.4	COMMENT CHANGER DE FICHIERS D'ENTREE/SORTIE .....	118
13.5	EXERCICES .....	122
14.	LE FILTRAGE (PREMIERE PARTIE) .....	124
14.1	PREMIERE VERSION D'UNE FONCTION DE FILTRAGE .....	125
14.2	EXERCICES .....	126
15.	RETOUR VERS LES LISTES .....	130
15.1	L'ORGANISATION DE LA MEMOIRE LISP .....	130
15.2	LA ZONE LISTE .....	131
15.3	LES FONCTIONS DE MODIFICATION DE LISTES .....	139
15.4	LES FONCTIONS EQ ET EQUAL .....	145
15.5	LES LISTES CIRCULAIRES .....	147
15.6	LES AFFECTATIONS .....	149
15.7	EXERCICES .....	152
16.	LES MACRO-FONCTIONS .....	154
16.1	EXERCICES .....	159
17.	LES DIVERSES FORMES DE REPETITION .....	160
17.1	APPEL DE FONCTION IMPLICITE (OU LE BRANCHEMENT) .....	160
17.2	LES ARGUMENTS FONCTIONNELS .....	161
17.3	QUELQUES ITERATEURS SUPPLEMENTAIRES .....	168
17.4	EXERCICES .....	171
18.	LE FILTRAGE (DEUXIEME PARTIE) .....	172
19.	SOLUTIONS AUX EXERCICES .....	191
19.1	CHAPITRE 1 .....	191

---

19.2	CHAPITRE 2.....	191
19.3	CHAPITRE 3.....	192
19.4	CHAPITRE 4.....	193
19.5	CHAPITRE 5.....	194
	19.5.1 chapitre 5.1.1.....	194
	19.5.2 chapitre 5.2.1.....	195
19.6	CHAPITRE 6.....	195
19.7	CHAPITRE 7.....	198
19.8	CHAPITRE 8.....	200
19.9	CHAPITRE 9.....	205
19.10	CHAPITRE 10.....	207
19.11	CHAPITRE 11.....	211
19.12	CHAPITRE 12.....	212
19.13	CHAPITRE 13.....	213
19.14	CHAPITRE 14.....	214
19.15	CHAPITRE 15.....	215
19.16	CHAPITRE 16.....	217
19.17	CHAPITRE 17.....	218
20.	BIBLIOGRAPHIE.....	219
21.	LES FONCTIONS LISP STANDARD.....	223
22.	INDEX.....	228

---