

2. LES FONCTIONS DE BASE : QUOTE, CAR, CDR, CONS

Les listes et les atomes sont donc les objets sur lesquels vous travaillez en LISP. Il existe des listes spéciales, nommées des *formes*, qui servent à indiquer à la machine-LISP qu'elle doit faire quelque chose. "Faire quelque chose" se dit, en LISP, *appeler une fonction*. Voyons dans l'ordre : d'abord la définition d'une forme :

$$\text{forme} ::= (\text{nom-de-fonction } \{ \text{argument}_1 \text{ argument}_2 \dots \text{argument}_n \})$$

Un *appel de fonction* ou une *évaluation d'une forme* ramène (ou *retourne*) une valeur. Toute fonction LISP ramène une valeur à l'appel.

Voici la définition d'une fonction TRES importante, la fonction **QUOTE** :

$$(\text{QUOTE } arg_1) \rightarrow arg_1$$

Elle sert à dire à la machine sur quel objet on veut travailler; elle ramène *en valeur* l'objet LISP donné en argument (de la fonction) tel quel. *Ramène en valeur* sera noté avec le signe " → ".

Quelques exemples d'appels de la fonction **QUOTE** :

(QUOTE TIENS)	→	TIENS
(QUOTE (A B C))	→	(A B C)
(QUOTE (THE (PUMPKIN (EATER))))	→	(THE (PUMPKIN (EATER)))

La fonction **QUOTE** est d'une utilité extrême dans la programmation en LISP : puisque la majorité des fonctions LISP évalue ses arguments, nous utilisons la fonction **QUOTE** pour les arguments que nous ne voulons pas évaluer. L'évaluation des arguments des fonctions peut ainsi être réalisée sans crainte : **QUOTE** nous ramène l'argument *tel quel*.

La fonction **QUOTE** est tellement importante et tellement souvent utilisée, qu'il existe une notation abrégée :

'TIENS	→	TIENS
'(A B C)	→	(A B C)
'(THE (PUMPKIN (EATER)))	→	(THE (PUMPKIN (EATER)))

Cette notation n'est qu'une abréviation : la machine, elle, comprend toujours la même chose, i.e. un appel de la fonction **QUOTE**. (La fonction **QUOTE** peut être comprise comme la fonction d'identité.)

Puisqu'on a des listes, et puisqu'on peut énumérer les éléments d'une liste, il existe des fonctions pour

accéder aux différents éléments d'une liste. D'abord la fonction **CAR** :¹

(CAR arg) → le premier élément de la liste *arg* donnée en argument.
arg doit obligatoirement être une liste

voici quelques exemples d'appels :

(CAR '(A B C)) → **A**
notez que l'argument a été QUOTE ! pourquoi ?
(CAR '(THE (PUMPKIN (EATER)))) → **THE**
(CAR '(((O C N H) P S))) → **((O C N H))**

naturellement, les fonctions LISP peuvent être composées :

(CAR (CAR '(((O C N H) P S)))) → **(O C N H)**
(CAR (CAR (CAR '(((O C N H) P S)))) → **O**

d'ailleurs :

(CAR '(CAR '(((O C N H) P S))) → **CAR**

Voyez-vous maintenant l'utilité de la fonction **QUOTE** ?

La fonction **CDR** peut être définie comme suit :

(CDR arg) → la liste *arg* donnée en argument sans le premier élément.
arg doit obligatoirement être une liste

CDR est la fonction complémentaire de la fonction **CAR**. Voici quelques exemples :

(CDR '(A B C)) → **(B C)**
(CDR '(THE (PUMPKIN (EATER)))) → **((PUMPKIN (EATER)))**
(CDR '(((O C N H) P S))) → **(P S)**

et en combinant des fonctions :

(CDR (CDR '(A B C))) → **(C)**
(CAR (CDR '(A B C))) → **B**
(CAR (CDR (CDR '(A B C)))) → **C**

Certaines combinaisons des fonctions **CAR** et **CDR** sont tellement utiles, qu'il en existe des écritures abrégées :

(CAR (CDR liste)) est la même chose que **(CADR liste)**
(CAR (CDR (CDR liste))) est la même chose que **(CADDR liste)**

La fonction **CADR** ramène le deuxième élément de sa liste argument, et la fonction **CADDR** ramène le troisième élément.

Vérifiez vous même qu'avec des combinaisons adéquates des fonctions **CAR** et **CDR** vous pouvez ramener

¹ Ce nom barbare ne subsiste que pour des raisons historiques : la première implémentation de LISP se faisait sur un ordinateur IBM-704. Un mot de cette machine était divisé en une partie "adresse" et une partie "décrement". Le Contenu de la partie Adresse d'un Registre livrait alors le **CAR** d'une liste et le Contenu de la partie Décrement d'un Registre livrait le **CDR** d'une liste. En honneur de cette première implémentation de LISP, toutes les versions suivantes ont gardé ces deux noms.

N'IMPORTE quel élément de N'IMPORTE quelle liste !

Jusqu'à présent nous connaissons une fonction ramenant son argument tel quel (**QUOTE**), une fonction qui ramène le premier élément d'une liste (**CAR**) et une (**CDR**) qui ramène le reste d'une liste, c'est-à-dire : la liste sans son premier élément. Il nous faut aussi une fonction pour CONStruire une nouvelle liste, c'est la fonction **CONS**, définie comme :

(**CONS** *argument₁* *liste*) → la liste *liste* avec la valeur de *argument₁* comme nouveau premier élément

exemples :

(CONS 'A '(B C))	→	(A B C)
(CONS '(A B) '(C D))	→	((A B) C D)
(CONS (CAR '(A B C))(CDR '(A B C)))	→	(A B C) ; !! ;
(CAR (CONS 'A '(B C)))	→	A
(CDR (CONS 'A '(B C)))	→	(B C)

Pour terminer cette première partie, voici une image d'une petite interaction avec la machine (La machine écrit un "?" quand elle attend que vous entrez quelque chose. Elle imprime la valeur de ce que vous avez demandé, précédée du signe "=", sur la ligne suivante.) :

```
? 'A
= A

? '(A B C)
= (A B C)

? ""(A B C)
= ""(A B C)

? (CAR '(A B C))
= A

? (CDR '(A B C))
= (B C)

? (CADR '(A B C))
= B

? (CADDR '(A B C))
= C

? (CONS 'A '(B C))
= (A B C)

? (CONS 'A (CONS 'B '()))
= (A B)

? (CONS (CAR '(A B C))(CDR '(A B C)))
= (A B C)
```

Remarque importante :

Par définition des fonctions **CAR** et **CDR**, le **CAR** de **NIL** est égal à **NIL**, et le **CDR** de **NIL** est également égal à **NIL**. Vous avez donc les relations suivantes :

$$\begin{array}{ll} (\text{CAR NIL}) & \rightarrow \text{NIL} \\ (\text{CAR } ()) & \rightarrow \text{NIL} \\ (\text{CDR NIL}) & \rightarrow \text{NIL} \\ (\text{CDR } ()) & \rightarrow \text{NIL} \end{array}$$

Remarquez également :

$$(\text{CONS } () \text{ '(A B C)}) = (\text{CONS NIL '(A B C)}) \rightarrow (()) \text{ A B C}$$

Bien évidemment, faire un **CONS** avec **NIL** et une liste revient à insérer une liste vide en tête de la liste donnée en deuxième argument. **NIL** en tant que deuxième argument d'un **CONS** correspond à mettre une paire de parenthèses autour du premier argument du **CONS** :

$$(\text{CONS 'A NIL}) = (\text{CONS 'A } ()) \rightarrow (\text{A})$$

2.1. EXERCICES

1. Donnez les résultats des appels des fonctions suivantes :

- a. $(\text{CAR '((A (B C)) D (E F)))} \rightarrow ?$
- b. $(\text{CDR '((A (B C)) D (E F)))} \rightarrow ?$
- c. $(\text{CADR (CAR '((A (B C)) D (E F)))}) \rightarrow ?$
- d. $(\text{CADDR '((A (B C)) D (E F)))} \rightarrow ?$
- e. $(\text{CONS 'NOBODY (CONS 'IS '(PERFECT)))} \rightarrow ?$
- f. $(\text{CONS (CAR '((CAR A) (CDR A))) (CAR '(((CONS A B))))}) \rightarrow ?$