

```
1 ;Samedi 6 Avril 2002 11:21:19 file : prt-hw.vlisp ;
2
3 ;----- pretty-print, par [pg];
4 ; transformation en evaluateur 24.3.02 (hw);
5
6 (setq --x-- careful careful nil)
7
8 (df pretty (l) pretty
9 (mapc l (lambda (f) (terpri) (pprint f))))
10
11 (de pprint (f ; x) pprint
12 (status print 3)
13 ; bit 0 : "" , bit 1 : pas d'espace.;
14 (setq lmargin 0)
15 (setq x (cdr (assoc (ftyp f) '((7 . de) (8 . df) (9 . macro)))))
16 (and x (interprete [x f . (fval f)]))
17 (terpri)
18 (status print 0)
19 f)
20
21 (de interprete (l ; x xx) interprete
22 (initializeInterprete)
23 (cond
24 ((null l) (casNil))
25 ((atom l) (casAtom l))
26 ((and (eq (car l) quote) (null (cddr l))) (casQuote))
27 ((and (listp (car l)) (eq (caar l) lambda)) (casLambda))
28 (t (initGeneralCase)
29 (let ((f (and (litatom (car l)) (get (car l) 'prettyFtn))))
30 (if f (apply f nil) (defaultCase)))
31 (exitGeneralCase))))
32
33 (defmacro initializeInterprete () initializeInterprete
34 '(status print 3) ; pour les appels externes ;
35
36 (defmacro casNil () casNil
37 '(princh "()")
38
39 (defmacro casAtom (l) casAtom
40 '(prin1 ,l)
41
42 (defmacro casQuote () casQuote
43 '(progn (princh "") (interprete (cadr l))))
44
45 (defmacro casLambda () casLambda
46 '(progn
47 (setq
48 l
49 ['let
50 (let ((v (cadar l)) (a (cdr l)))
51 (if (null v) nil [(nextl v) (nextl a)] . (self v a)))
52 . (cddar l))
53 (interprete l))
```

```

54
55 (defmacro initGeneralCase ()                               initGeneralCase
56   '(progn (setq xx (outpos)) (princh "")))
57
58 (defmacro exitGeneralCase ()                               exitGeneralCase
59   '(progn (and 1 (princh " . ") (princh 1)) (princh "")))
60
61 (defmacro defaultCase ()                                   defaultCase
62   '(progn (t+3) (interprete (nextl 1)) (while (listp 1) (p-p1)) (t-3)))
63
64 (defmacro p-p1 ()                                         p-p1
65   '(progn (princh " ") (interprete (nextl 1))))
66
67 (defmacro t+3 ()                                          t+3
68   '(setq lmargin (+ lmargin 3)))
69
70 (defmacro t-3 ()                                          t-3
71   '(setq lmargin (- lmargin 3)))
72
73 (defmacro p-progn (?)                                      p-progn
74
75   '(if (and (null (cdr 1)) (null ,?))
76       (p-p1)
77       ; un seul argument;
78       (t+3)
79       ; plusieurs;
80       (while (listp 1)
81         (if (> lmargin (outpos)) (outpos lmargin) (terpri))
82         (interprete (nextl 1)))
83       (t-3)))
84
85 (defmacro p-cond ()                                       p-cond
86   '(progn
87     (t+3)
88     (while (listp 1)
89       (terpri)
90       (princh "")
91       (let (l (nextl 1)) (interprete (nextl 1)) (if 1 (p-progn)))
92       (princh ""))
93     (t-3)))
94
95 (mapc '(progn ; type progn; prog1 and exit or)
96       (lambda (x) (put x 'prettyFtn '(lambda () (interprete (nextl 1)) (p-progn)))))
97
98 (mapc '(lambda ; type while; escape ifn let mapc mapcar while until)
99       (lambda (x) (put x 'prettyFtn '(lambda () (interprete (nextl 1)) (p-p1) (p-progn t)))))
100
101 (mapc '(de ; type def; df dm dmc)
102       (lambda (x)
103         (put x 'prettyFtn '(lambda () (interprete (nextl 1)) (p-p1) (p-p1) (p-progn t)))))
104
105 (mapc '(cond ; type cond;)
106       (lambda (x) (put x 'prettyFtn '(lambda () (interprete (nextl 1)) (p-cond)))))
107
108 (mapc '(selectq ; type selectq;)
109       (lambda (x) (put x 'prettyFtn '(lambda () (interprete (nextl 1)) (p-p1) (p-cond)))))

```

```
110  
111 (setq careful --x--)  
112  
113
```

Cross Reference

casAtom defmacro 39
casLambda defmacro 45
casNil defmacro 36
casQuote defmacro 42
defaultCase defmacro 61
exitGeneralCase defmacro 58
initGeneralCase defmacro 55
initializeInterprete defmacro 33
interprete de 21
p-cond defmacro 85
p-p1 defmacro 64
p-progn defmacro 73
pprint de 11
pretty df 8
t+3 defmacro 67
t-3 defmacro 70

1	--x--	6 111
2	?	73 75
3	a	50 51 51
4	casAtom	25 39
5	casLambda	27 45
6	casNil	24 36
7	casQuote	26 42
8	defaultCase	30 61
9	defmacro	33 36 39 42 45 55 58 61 64 67 70 73 85
10	exitGeneralCase	31 58
11	f	9 9 11 15 16 16 19 29 30 30
12	initGeneralCase	28 55
13	initializeInterprete	22 33
14	interprete	16 21 43 53 62 65 82 91 96 99 103 106 109

15	l	8 9 21 24 25 25 26 26 27 27 29 29 39 40 43 48 50 50 52 53 59 59 62 62 65 75 80 82 88 91 91 91 91 96 99 103 106 109
16	macro	15
17	p-cond	85 106 109
18	p-p1	62 64 76 99 103 103 109
19	p-progn	73 91 96 99 103
20	pprint	9 11
21	pretty	8
22	prettyFtn	29 96 99 103 106 109
23	t+3	62 67 78 87
24	t-3	62 70 83 93
25	v	50 51 51 51
26	x	11 15 16 16 21 96 96 99 99 102 103 106 106 109 109
27	xx	21 56

;Samedi 6 Avril 2002 11:21:19 end of file : prt-hw.vlisp ;