

BOOLEANS AND CONSTANTS

T	NIL
true	Represents false or empty list

CONTROL FLOW

IF	WHEN	COND
Executes <i>then</i> if true Otherwise <i>else</i> (if condition then [else])	Like <i>if</i> , but only has a true branch, we can run multiple expressions	Like a chain of if/else if

LOGICAL OPERATORS

NOT	AND	OR	EQUAL
(NOT x) returns T if x is false, else NIL	(AND x y ...) returns NIL at the first false value, else the last true one	(OR x y ...) returns the first true value, else NIL	(EQUAL a b) returns T if a and b are equal

LIST CONSTRUCTORS

CONST
(cons a b) Constructs a pair from a and b. (cons 1 '(2 3)) ; => (1 2 3)

LIST ACCESSORIES

CAR/FIRST	CDR	SECOND	LENGTH
Returns first element	Gets the rest of the list excluding the first element	Gets the second element of the list	Returns the number of elements in a list

FUNCTION DEF. AND SCOPE

DEFUN	LABELS	LET	LET*
(defun name (args) body...) defines a function	Defines local recursive functions (labels ((name (args)	Creates local variables (let ((var val) ...)	Like let but allows each variable to depend on the

	body) ...) main-body...)	body...)	previous one
--	-----------------------------	----------	--------------

FUNCTION CALLING

FUNCALL	QUOTE or ‘
Calls a function passes as a value	Prevents evaluation. Example; ‘(1,2,3) is a literal list.

ARITHMETIC AND COMPARISON

Basic arithmetic	Numeric comparisons								
<table><tr><td>+</td><td>-</td><td>*</td><td>/</td></tr></table>	+	-	*	/	<table><tr><td><</td><td><=</td><td>>=</td><td>=</td></tr></table>	<	<=	>=	=
+	-	*	/						
<	<=	>=	=						