

SYMBOL TABLE for keywords.lusd

Total Tokenized Lexemes :63
Elapsed Time :0.0

LINE	LEXEME	TOKEN
1	all	KEYWORD_ALL
2	and	KEYWORD_AND
3	ask	KEYWORD_ASK
4	bool	KEYWORD_BOOL
5	break	KEYWORD_BREAK
6	catch	IDENTIFIER
7	char	KEYWORD_CHAR
8	check	IDENTIFIER
9	class	KEYWORD_CLASS
10	const	KEYWORD_CONST
11	continue	KEYWORD_CONTINUE
12	display	KEYWORD_DISPLAY
13	do	KEYWORD_DO
14	double	KEYWORD_DOUBLE
15	elif	KEYWORD_ELIF
16	else	KEYWORD_ELSE
17	false	KEYWORD_FALSE
18	finally	IDENTIFIER
19	float	KEYWORD_FLOAT
20	for	KEYWORD_FOR
21	foreach	KEYWORD_FOREACH
22	from	KEYWORD_FROM
23	func	KEYWORD_FUNC
24	if	KEYWORD_IF
25	import	KEYWORD_IMPORT
26	info	KEYWORD_INFO
27	init	KEYWORD_INIT

LINE	LEXEME	TOKEN
28	inheritall	KEYWORD_INHERITALL
29	in	KEYWORD_IN
30	int	KEYWORD_INT
31	list	KEYWORD_LIST
32	main	KEYWORD_MAIN
33	not	KEYWORD_NOT
34	null	KEYWORD_NULL
35	or	KEYWORD_OR
36	obj	KEYWORD_OBJ
37	private	KEYWORD_PRIVATE
38	protected	KEYWORD_PROTECTED
39	public	KEYWORD_PUBLIC
40	quit	KEYWORD_QUIT
41	raise	IDENTIFIER
42	repeat	KEYWORD_REPEAT
43	return	KEYWORD_RETURN
44	str	KEYWORD_STR
45	then	KEYWORD_THEN
46	this	KEYWORD_THIS
47	true	KEYWORD_TRUE
48	try	IDENTIFIER
49	until	KEYWORD_UNTIL
50	while	KEYWORD_WHILE
51	3	INT_LITERAL
51	+	OP_ADDITION
51	-	OP_SUBTRACTION
51	++	OP_INCREMENT
51	1	INT_LITERAL
51	/	OP_DIVISION
51	4	INT_LITERAL

LINE	LEXEME	TOKEN
51	%	OP_MODULO
51	2	INT_LITERAL
51	+=	OP_ADDITION_ASSIGNMENT
51	5	INT_LITERAL
51	+	OP_ADDITION
51	/	OP_DIVISION