

SYMBOL TABLE for test.lusd

Total Tokenized Lexemes :218

Elapsed Time :0.007002115249633789

LINE	LEXEME	TOKEN
1	import	KEYWORD_IMPORT
1	TestLibrary	IDENTIFIER
1	;	STMT_TERMINATOR
3	public	KEYWORD_PUBLIC
3	class	KEYWORD_CLASS
3	testClass	IDENTIFIER
3	:	CODEBLK_INDICATOR
4	{	OPEN_CURLY_BRACKET
5	init	KEYWORD_INIT
5	(OPEN_PARENTHESIS
5	this	KEYWORD_THIS
5	,	SEPARATOR
5	int	KEYWORD_INT
5	num	IDENTIFIER
5)	CLOSE_PARENTHESIS
5	:	CODEBLK_INDICATOR
6	{	OPEN_CURLY_BRACKET
7	this	KEYWORD_THIS
7	.	OBJECT_DELIMITER
7	property_2_1	IDENTIFIER
7	=	OP_ASSIGNMENT
7	.5	DOUBLE_LITERAL
7	;	STMT_TERMINATOR
8	display	KEYWORD_DISPLAY
8	(OPEN_PARENTHESIS
8	num	IDENTIFIER
8)	CLOSE_PARENTHESIS

LINE	LEXEME	TOKEN
8	;	STMT_TERMINATOR
9	this	KEYWORD_THIS
9	.	OBJECT_DELIMITER
9	property_2_1	IDENTIFIER
9	;	STMT_TERMINATOR
10	}	CLOSE_CURLY_BRACKET
12	public	KEYWORD_PUBLIC
12	func	KEYWORD_FUNC
12	TestMethod	IDENTIFIER
12	(OPEN_PARENTHESIS
12)	CLOSE_PARENTHESIS
12	:	CODEBLK_INDICATOR
13	{	OPEN_CURLY_BRACKET
14	display	KEYWORD_DISPLAY
14	(OPEN_PARENTHESIS
14	"Hi Word"	STRING_LITERAL
14)	CLOSE_PARENTHESIS
14	;	STMT_TERMINATOR
15	}	CLOSE_CURLY_BRACKET
17	public	KEYWORD_PUBLIC
17	func	KEYWORD_FUNC
17	TestMethod3	IDENTIFIER
17	(OPEN_PARENTHESIS
17	str	KEYWORD_STR
17	val1	IDENTIFIER
17)	CLOSE_PARENTHESIS
17	:	CODEBLK_INDICATOR
18	{	OPEN_CURLY_BRACKET
19	display	KEYWORD_DISPLAY
19	(OPEN_PARENTHESIS

LINE	LEXEME	TOKEN
19	val1	IDENTIFIER
19)	CLOSE_PARENTHESIS
19	;	STMT_TERMINATOR
19	display	KEYWORD_DISPLAY
19	(OPEN_PARENTHESIS
19	val1	IDENTIFIER
19)	CLOSE_PARENTHESIS
19	;	STMT_TERMINATOR
20	}	CLOSE_CURLY_BRACKET
21	}	CLOSE_CURLY_BRACKET
23	main	KEYWORD_MAIN
23	(OPEN_PARENTHESIS
23)	CLOSE_PARENTHESIS
23	{	OPEN_CURLY_BRACKET
24	display	KEYWORD_DISPLAY
24	(OPEN_PARENTHESIS
24	"Hello Word!\b"	STRING_LITERAL
24)	CLOSE_PARENTHESIS
24	i	IDENTIFIER
24	+++++	UNKNOWN_TOKEN
24	;	STMT_TERMINATOR
25	integer	KEYWORD_INT
25	intvar	IDENTIFIER
25	=	OP_ASSIGNMENT
25	3	INT_LITERAL
25	+	OP_ADDITION
25	4	INT_LITERAL
25	;	STMT_TERMINATOR
26	float	KEYWORD_FLOAT
26	FloatVar	IDENTIFIER

LINE	LEXEME	TOKEN
26	=	OP_ASSIGNMENT
26	420.0f	FLOAT_LITERAL
26	;	STMT_TERMINATOR
27	double	KEYWORD_DOUBLE
27	double_var	IDENTIFIER
27	=	OP_ASSIGNMENT
27	0.69	DOUBLE_LITERAL
27	;	STMT_TERMINATOR
29	char	KEYWORD_CHAR
29	char_Va_r	IDENTIFIER
29	=	OP_ASSIGNMENT
29	'd'	CHAR_LITERAL
29	;	STMT_TERMINATOR
30	str	KEYWORD_STR
30	stringVariable	IDENTIFIER
30	=	OP_ASSIGNMENT
30	"Nico Nico niiiii\n"	STRING_LITERAL
30	;	STMT_TERMINATOR
31	list	KEYWORD_LIST
31	Ex_Taylor	IDENTIFIER
31	[OPEN_SQUARE_BRACKET
31	3	INT_LITERAL
31]	CLOSE_SQUARE_BRACKET
31	=	OP_ASSIGNMENT
31	[OPEN_SQUARE_BRACKET
31	"Loki"	STRING_LITERAL
31	,	SEPARATOR
31	"What happen bella"	STRING_LITERAL
31	,	SEPARATOR
31	"Red Scarf"	STRING_LITERAL

LINE	LEXEME	TOKEN
31]	CLOSE_SQUARE_BRACKET
31	;	STMT_TERMINATOR
33	str	KEYWORD_STR
33	9d0fvm_3d	UNKNOWN_TOKEN
33	=	OP_ASSIGNMENT
33	"Test"	STRING_LITERAL
33	;	STMT_TERMINATOR
34	char	KEYWORD_CHAR
34	__Dienamayt	UNKNOWN_TOKEN
34	=	OP_ASSIGNMENT
34	'dc'	UNKNOWN_TOKEN
34	;	STMT_TERMINATOR
35	str	KEYWORD_STR
35	Inabikari_sunawachi_eien_nari_	UNKNOWN_TOKEN
35	;	STMT_TERMINATOR
37	str	KEYWORD_STR
37	order_var	IDENTIFIER
37	=	OP_ASSIGNMENT
37	"Oratrice Mecanique d'Analyse Cardinale;	UNKNOWN_TOKEN
39	if	KEYWORD_IF
39	(OPEN_PARENTHESIS
39	intvar	IDENTIFIER
39	<=	LESS_OR_EQUAL_OP
39	3	INT_LITERAL
39	and	KEYWORD_AND
39	not	KEYWORD_NOT
39	5	INT_LITERAL
39)	CLOSE_PARENTHESIS
39	then	KEYWORD_THEN

LINE	LEXEME	TOKEN
39	:	CODEBLK_INDICATOR
40	{	OPEN_CURLY_BRACKET
41	intvar	IDENTIFIER
41	=	OP_ASSIGNMENT
41	5	INT_LITERAL
41	+	OP_ADDITION
41	3	INT_LITERAL
41	+	OP_ADDITION
41	9	INT_LITERAL
41	+	OP_ADDITION
41	7	INT_LITERAL
41	/	OP_DIVISION
41	2	INT_LITERAL
41	;	STMT_TERMINATOR
42	}	CLOSE_CURLY_BRACKET
43	elif	KEYWORD_ELIF
43	(OPEN_PARENTHESIS
43	intvar	IDENTIFIER
43	!=	INEQUALITY_OP
43	3	INT_LITERAL
43)	CLOSE_PARENTHESIS
43	:	CODEBLK_INDICATOR
44	{	OPEN_CURLY_BRACKET
45	intvar	IDENTIFIER
45	=	OP_ASSIGNMENT
45	10	INT_LITERAL
45	;	STMT_TERMINATOR
46	}	CLOSE_CURLY_BRACKET
47	else	KEYWORD_ELSE
47	:	CODEBLK_INDICATOR

LINE	LEXEME	TOKEN
48	{	OPEN_CURLY_BRACKET
49	pass	KEYWORD_PASS
49	;	STMT_TERMINATOR
50	}	CLOSE_CURLY_BRACKET
52	for	KEYWORD_FOR
52	(OPEN_PARENTHESIS
52	int	KEYWORD_INT
52	i	IDENTIFIER
52	=	OP_ASSIGNMENT
52	0	INT_LITERAL
52	;	STMT_TERMINATOR
52	i	IDENTIFIER
52	<	LESS_THAN_OP
52	5	INT_LITERAL
52	;	STMT_TERMINATOR
52	i	IDENTIFIER
52	++	OP_INCREMENT
52)	CLOSE_PARENTHESIS
52	:	CODEBLK_INDICATOR
53	{	OPEN_CURLY_BRACKET
54	display	KEYWORD_DISPLAY
54	(OPEN_PARENTHESIS
54	testClass	IDENTIFIER
54	.	OBJECT_DELIMITER
54	property_2_1	IDENTIFIER
54)	CLOSE_PARENTHESIS
54	;	STMT_TERMINATOR
55	}	CLOSE_CURLY_BRACKET
57	while	KEYWORD_WHILE
57	(OPEN_PARENTHESIS

LINE	LEXEME	TOKEN
57	int	KEYWORD_INT
57	j	IDENTIFIER
57	<	LESS_THAN_OP
57	5	INT_LITERAL
57)	CLOSE_PARENTHESIS
57	:	CODEBLK_INDICATOR
58	{	OPEN_CURLY_BRACKET
59	j	IDENTIFIER
59	++	OP_INCREMENT
59	;	STMT_TERMINATOR
60	}	CLOSE_CURLY_BRACKET
61	}	CLOSE_CURLY_BRACKET