

comment.line
comment.block

punctuation.definition.comment

punctuation.definition.block.autolispex

entity.name.function.autolispex

constant.language

entity.name.variable

string.quoted.double.autolispex

keyword.control.conditional.autolispex

keyword.control.logical.autolispex

keyword.operator.other.autolispex

keyword.arithmetic.autolispex

constant.numeric.autolispex

punctuation.definition.quote.autolispex

constant.language.equality

entity.name.class

entity.name.variable.parameter

```
;; for testing syntax analysis
(defun abc ()
  (setq a 123)
  (setq b (1+ a) b1 PI B2 PAUSE)
  PI
  PAUSE
  PI
  (setq c (+ 1 2))
  (setq d *error* (d1 vl-catch-all-error-p d))
  (setq e 'abc)
  (setq f "string\n\124 aa\\a")
  (getvar "Filletr<d")
  (setq f (if (> 1 2) (>= 1 100) (= 1 1) )))
  (setq g ~ 100))
  (setq h (cond (if (eq 1 1) )))
  (setq i (abs 100) j (cos 45))
  (setq k T 1 nil)
  (princ a)
  (apply + (1 2 3))
  (T T)
  (nil nil)
  (t)
  (Nil)
  (exit)
  T
  nil
)
(defun c:xyz ( zzz / yyy )
  (princ "xyz")
  (c:xyz)
)
```

keyword.operator.arithmetic.systemkeyword

keyword.error.language

constant.character.escape.language.autolispex

support.other.language

keyword.operator.logical.autolispex

punctuation.definition.parameter.divider.autolispex