

# 求值

aborn

2018-05-10

## Contents

<b>1</b>	<b>求值</b>	<b>1</b>
<b>2</b>	<b>表达式类型</b>	<b>1</b>
2.1	自解释表达式 . . . . .	1
2.2	标识符类型 . . . . .	1
2.3	自动载入 (Autoloading) . . . . .	2
<b>3</b>	<b>引用</b>	<b>2</b>
<b>4</b>	<b>反引号</b>	<b>2</b>

---

## 1 求值

Lisp 解释器会对表达式进行求值操作,也可以手工调用求值方法 **eval**。Lisp 解释器通常先读取 Lisp 表达式,然后对表达式进行求值。其实,读取和求值是两个相互独立的过程,它们也可以进行单独操作。

## 2 表达式类型

表达式是一种用于求值的 lisp 对象, Emacs 有三种不同的求值表达式类型: 标识符 (Symbols)、列表和其他类型。下面从其他类型开始介绍。

### 2.1 自解释表达式

自解释类型,其意思很明确是自己对自己求值,例如 25 自解释成 25,自字符串"foo" 自解释成"foo"。

## 2.2 标识符类型

当标识符类型被求值，它将被当成变量使用，求值的结果就是变量的值。如果变量没有值，Lisp 解释器会抛出一个错误提示。

```
(setq a 123)    ;; 123
(eval 'a)       ;; 123
a               ;; 123
```

## 2.3 自动载入 (Autoloading)

自动载入的特性允许函数或者宏还没有载入到 Emacs 中前使用它们。

## 3 引用

引用 (quote) 是一种特殊表达式，它返回它的参数且不对其进行求值。它提供了一种在程序里包含标识符常量和列表却不需要对其求值的使用方式。

```
(quote object)
```

它返回 object，但不对 object 进行求值操作。它提供了一种简写方式，即“”，'object。

## 4 反引号

反引号 (backquote ‘) 可用于列表，它与引用唯一的区别的，它允许对列表中部分元素进行求值。采用逗号 (,) 来标识那些元素需要进行求值，下面是一些例子：

```
`(a list of ,(+ 2 3) elements)    ;; (a list of 5 elements)
`(1 2 (3 ,(+ 4 5)))              ;; (1 2 (3 9))
```