

宏 Macros

预先定义的变量与方法.

与调用函数不同，宏执行的并不是简单的调用参数，而是把对宏的调用更改为替换后的主体的实际文本.

- 在 C/C++ 中宏的定义通过 `#define` 实现的.
- 在 Scheme 中宏通过 `define-macro` 实现. `define-macro` 的使用方法类似，但它“返回”一个表达式（而非表达式的值）. 这可以用 `quasiquote` 来实现.

例如 (`define-macro (unless cond body) '(if (not ,cond) ,body)`)

在程序中的所有 (`unless cond body`) 都会被替换为 (`(if (not cond) body)`)

`quasiquote` 表示在 `quote` 的基础上计算前面带 `'` 的表达式（例如上例中的 `cond` 和 `body`）。如果用函数实现 `unless`，那么 `body` 一定会被执行，可能会引发不必要的错误。

另一个例子是实现列表递推式。

(`define-macro (for-list var expr lst)`)

‘(`map (lambda (,var) ,expr) ,lst`)’

计算 (`for-list (x) (* xx) '(1 2 3 4 5)`)

相当于计算 (`map (lambda (x) (* xx)) '(1 2 3 4 5)`)