



@红烧LO (何婧誉)

---

大家一起来 (不)  
写宏

我是。。。

- ▶ 剑桥大学计算机系本科毕业
- ▶ 工作7年，其中和Clojure为伴5年
- ▶ 写过网站、股票交易平台、分布式风险系统、语模生成系统、语模质监系统
- ▶ 现在还是主要Clojure，附加着写点Hadoop/Spark
- ▶ 很不学术，不爱用术语，还不会写Haskell（

如果你这场TALK里什么都不记得，那请至少记得以下这条：

---

除非必要，别写宏

还没被逼疯的Lisp前辈们

宏是什么？能吃吗？

# 宏的定义

- ▶ 我喜欢简单地定义成“生成代码的代码”
- ▶ 吃进去的是代码，吐出来的也是代码
- ▶ 那可以有生成宏的宏吗？
- ▶ 可以呀，那就是“生成生成代码的代码的代码”
- ▶ 生成生成生成代码的代码的代码。。。
- ▶ 子子孙孙无穷尽也。。。 （极限≈你的脑容量）



在LISP表达式的一生中，  
有三个最重要的时刻：  
读取期，编译期和运行期。

Paul Graham

今天要讲的不是。。

---

# 改变语法的宏

- ▶ 首先，有一种宏叫读取器宏（reader macro）
- ▶ 顾名思义，此宏在读取期发生，可以指挥读取器做不同往常的事情。
- ▶ 主要作用是？改变语法。
- ▶ 历史上Lisp产生的社会问题（贺师俊）的罪魁祸首。
- ▶ 每个人写出了不同的语言，谁都看不懂（惯）对方的Lisp。
- ▶ Clojure有读取器宏（`'`等等），万幸的是使用语言的人不能写这种宏，因此能写的Clojure宏不能改变语法。

今天要讲的是。。。

---

## CLOJURE的宏

- ▶ 不能改变语法
- ▶ 可以生成代码
- ▶ 使用backquote大法
- ▶ macroexpand查查
- ▶ 搞定啦！
- ▶ 举个🌰



## QUOTE(')

- ▶ clojure最基本要用到的东西
- ▶ quote一个表达式，使其在现阶段中不被eval
- ▶ 这就是我们刚才想要不eval b的话必须要用的法宝
- ▶ 举个🌰：



## SYNTAX-QUOTE()和UNQUOTE(~)

- ▶ 之前讲了，光用list和'可以写宏，但是写出来的宏又臭又长，太难看了
- ▶ 没关系，Clojure有syntax-quote
- ▶ syntax-quote干的事情就是把整个form里面的东西全部一次性quote起来，也就是不用写list，也不用一个一个单独来'
- ▶ 那我form里不想quote的东西怎么办？简单-用unquote (~)
- ▶ 举个🌰

## GEN-SYM(#)

- ▶ 这个可能是最难解释的读取器宏
- ▶ 在我们之前的栗子中，外面有一个`result`，里面也有一个`result`的话，不能两全。我们当然可以说里面的不要叫`result`，但是现实中你有时候不能知道外面所有的`state`都叫啥
- ▶ 为了防止这个问题，Clojure强制你在`macro`的`let`里使用`#`。`#`的作用就是生成那一堆乱七八糟的名字（比如`result__142__auto__`），以防止出现重复。而且这样，如果你想用外面的`result`，就直接用`result`就好了。

你以为这就完了？并没有。。。因为。。。

- ▶ 还有~@，学名是unquote-splice
- ▶ 作用是unquote+把list里面的东西一个个拿出来
- ▶ 主要用于宏吃进一段代码的时候
- ▶ 很难讲得清楚
- ▶ 再来个🌰吧

你们现在应该会写宏了吧！来，重要的事情再说一遍

---

除非必要，别写宏

还没被逼疯的Lisp前辈们

## 什么时候是必要写宏的时候呢？

- ▶ 就是函数已经不能满足你需要的功能的时候
- ▶ 确实很少发生
- ▶ 在我的Clojure工作生涯里，只碰到过两次需要写宏的时候
- ▶ 第一次是动态加载java库，并使用宏来call该库的API，因为如果用函数的话在java库起始缺失的情况下，不能编译
- ▶ 第二次是java库新版本改变了API，而我的项目需要即时根据该库的版本同时满足两版API的需求，那是我唯一一次需要写二阶宏
- ▶ 两次都是必须要在编译期生成代码（后面那次是生成再被编译一次的代码）

## 二阶宏

```
(defmacro defmultiv
  [fn-name fn-params & version-bodies]
  (let [version-body-pairs (partition 2 version-bodies)]
    `(defmacro ~fn-name
      ~fn-params
      (cond
        ~@(mapcat
              (fn [[v# b#]] (list `(sdk_>= ~v#) b#))
              version-body-pairs))))))

(defmultiv get-capitalization-hint
  [hint-key]
  :3.0.0.195
  `({:lower      ResultsFilter$CapitalizationHint/DEFAULT
     :initial-upper ResultsFilter$CapitalizationHint/INITIAL_UPPER_CASE
     :upper       ResultsFilter$CapitalizationHint/UPPER_CASE} ~hint-key)
  :1.6
  `({:lower      ResultsFilter$CapitalizationHint/LOWER_CASE
     :initial-upper ResultsFilter$CapitalizationHint/INITIAL_UPPER_CASE
     :upper       ResultsFilter$CapitalizationHint/UPPER_CASE} ~hint-key))
```

- ▶ 呐，如果不需要，真的别写这样的东西
- ▶ 写了的话，也别再碰它了。。。。

除非必要，别写宏

还没被逼疯的Lisp前辈们



# ANY QUESTIONS?

别叫我现场写宏， 我选择死亡