深入浅出Clojure宏

刘家财
http://liujiacai.net/

2017-12-23

大纲

- Why
  - DSL
- What
  - Function run at compile-time
  - Code as Data
- 实战
  - Rules of thumbs 经验法则
  - Quote/syntax-quote/unquote/unquote-splicing
  - 宏“卫生”(Hygiene)
  - Macro-writing macro

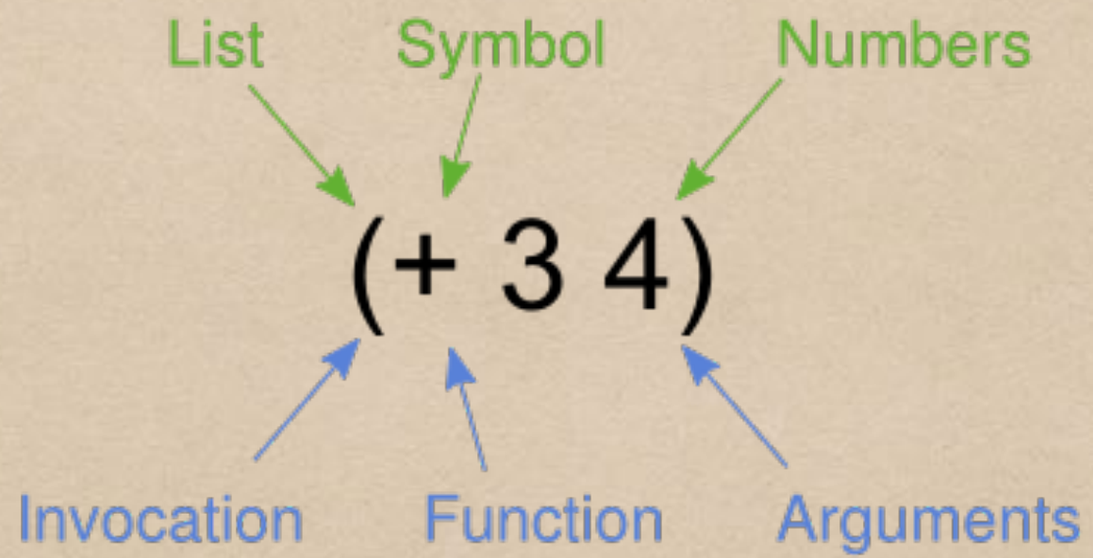
Why Macro?

- 专为 Lisp 程序员提供的写 DSL 的武器
  - 同步方式写异步代码的 core.async
  - Ring 路由库 Compojure
- Code-writing code 简化代码
- Very funny

What is Macro

- 编译时期运行的函数
  - 输入 code，输出 code

Code as Data



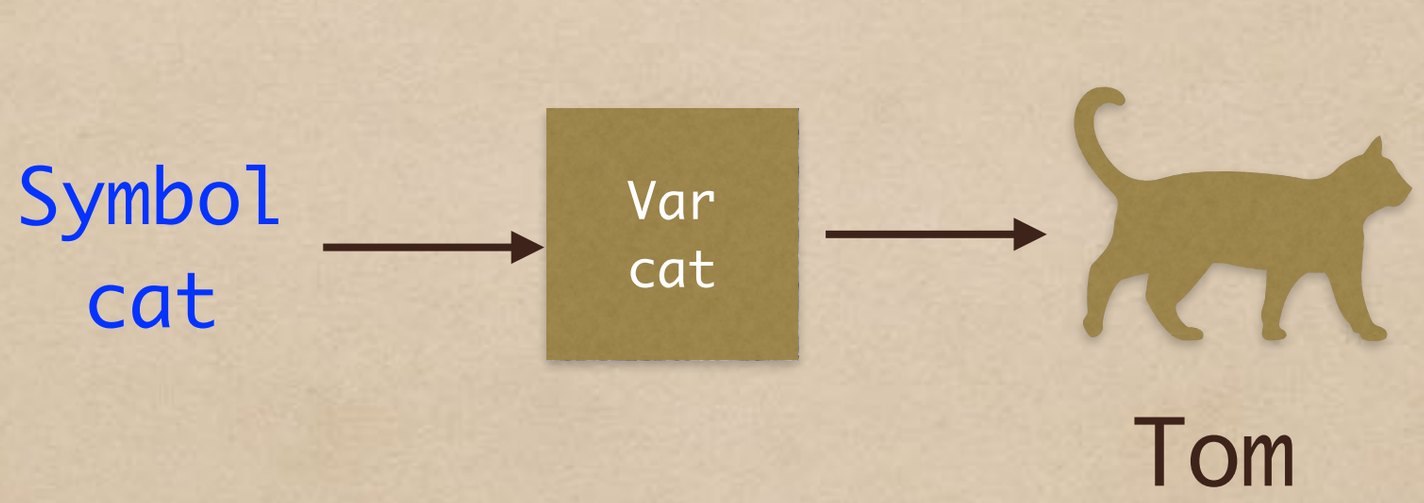
Code as Data

- 在 Clojure 里面，大部分类型字面量，其求值结果是其自身。有两个例外：
  - symbol，代指其他值
  - list，表示函数调用

```
;; Numeric types
42              ; Long - 64-bit integer (from -2^63 to 2^63-1)
6.022e23        ; Double - double-precision 64-bit floating point
42N             ; BigInt - arbitrary precision integer
1.0M            ; BigDecimal - arbitrary precision fixed-point decimal
22/7            ; Ratio
;; Character types
"hello"         ; String
\e              ; Character
;; Other types
nil             ; null value
true            ; Boolean (also, false)
#"[0-9]+"       ; Regular expression
:alpha          ; Keyword
:release/alpha  ; Keyword with namespace
map             ; Symbol
+               ; Symbol - most punctuation allowed
clojure.core/+  ; Namespaced symbol
;; Collection types
'(1 2 3)     ; list
[1 2 3]      ; vector
#{1 2 3}     ; set
{:a 1, :b 2} ; map
```

Code as Data

```
(def cat "Tom")
```




```
user> (resolve 'cat)
#'user/cat
```

实战-Rules of Thumbs

1. 如果函数能完成相应功能，不要写宏。在需要构造语法抽象（比如when）或新的DSL 时再去用宏
2. 写一个宏使用的 demo，并手动展开
3. 使用macroexpand, macroexpand-1 与 clojure.walk/macroexpand-all 去验证宏是如何工作的
4. 在 REPL 中测试
5. 如果一个宏比较复杂，尽可能拆分成多个函数

注意事项

- 内部 bindings 时，使用 gensym 保证宏“卫生”
- 不要暴露宏实现细节
  - 保证宏对参数不会多次求值，避免副作用
- macroexpand is your friend!

资料

- 示例代码
- 由浅入深学习 Lisp 宏之理论篇
- 由浅入深学习 Lisp 宏之实战篇
- 书📖《Mastering Clojure Macros》

祝大家能找到一门
受用终身的语言 / 项目
享受编程的乐趣
体味 Clojure

群名称：SICP读书群
群　号：119845407

公众号