



LAMBDA LOUNGE

## Macros in Common Lisp

**Christopher Mark Gore**

<http://www.cgore.com>

[cgore@cgore.com](mailto:cgore@cgore.com)

 @cgore

**Thursday, May 1, AD 2014**

## Getting Started with Common Lisp

1. Install Linux.

`http://aptosid.com`

2. Install SBCL and some libraries.

```
apt-get install sbcl{,-doc,-source} \  
cl-{asdf,cffi}
```

3. Install Emacs and SLIME *(Not strictly required.)*

```
apt-get install emacs{,-goodies-el} cl-swank \  
cl-swank slime common-lisp-controller
```

## **Lisp Macros Are Vastly Superior To:**

1. Not having macros at all.
2. Vim/Emacs<sup>a</sup> macros.
3. Microsoft Word/Excel macros.
4. C pre-processor macros.
5. Scheme “hygienic” macros.

---

<sup>a</sup>Excluding `defmacro` in Elisp, which is equivalent.

**Why Not Just Functions?**

**TO DO**

# **TO DO**

## **How Macros Work**

**TO DO**

**Backquote**

**TO DO**

**Gensym**

**Macro Expansion**

**TO DO**



**Macros are Programs Too**

**TO DO**

**TO DO** **WITH-\* Macros**

## **Deconstructing Parameter Lists**

**TO DO**

## **Anaphoric Macros**

**TO DO**

## **Compile Time and Run Time**

In Lisp, this distinction still exists, but it doesn't work at all like you are used to.

**TO DO**

## **Redefining Macros**

**TO DO**

**TO DO**

**Symbol Macros**

**TO DO** **DSLs via Macros**



**Growing the Language Itself via Macros**

**TO DO**

**Ancient History: F-Expressions**

**TO DO**

## **Reader Macros**

These are really cool and complicated. You can completely redefine the language with these, making your own language, even one with syntax (Ruby, C, C++, etc.)

Perhaps another day.

***Questions?***