

Homebrew



“The missing package manager for OS X”

Aaron Feng

Who Am I?

- Orangizer for: Philly Lambda, Philly AWS, Hackathons (Center City)
- Work: Algorithmics (Financial Risk)
- Languages (professionally): C#, Java, Ruby, Clojure, Javascript

Package manager should be....

- Self-contained
- Not require sudo, **EVER!**
- Easy to use
- Lots of new packages
- Ability to create new packages and redistribute



Sudo can be dangerous

Before we talk about
homebrew, let's take a
detour of how I used to
manage my packages

I am Sure You Have...

- Purchased a new machine
- Cleanly installed OS without upgrading
- Experienced a hardware failure
- Broken your OS during a drunken coding session

I like everything installed
in a predefined location,
and not associated with
my base OS install

Wait, How did I setup my dev env again?
Download X from where? Compile Y?
Use binary of Z?
3 days later.....



This happened to me way
too many times, so I
started to write shell
scripts that manage my
packages

```

if [ ! -f $INSTALLED/ghc-6.10.4 ]
then
    echo -e "\033[32m Downloading ghc-6.10.4 to $DOWNLOAD \033[0m"
    test -f $DOWNLOAD/ghc-6.10.4.tar.bz2 || curl -C - -o $DOWNLOAD/ghc-6.10.4.tar.bz2 -O http://www.haskell.org/ghc/dist/6.10.4/maeder/ghc-6.10.4-powerpc-apple
    test -d $DOWNLOAD/ghc-6.10.4 || tar -xvf $DOWNLOAD/ghc-6.10.4.tar.bz2 -C $DOWNLOAD

    echo -e "\033[36m Compiling and Installing ghc-6.10.4 in $USR \033[0m"
    cd $DOWNLOAD/ghc-6.10.4
    ./configure --prefix=$USR/ghc-6.10.4
    make
    make install
    cd -
    echo -e "\033[36m Installed ghc-6.10.4 sucessfully \033[0m"
    echo `date` >> $INSTALLED/ghc-6.10.4
else
    echo -e "\033[1m ghc-6.10.4 is already installed - skipping \033[0m"
fi

if [ ! -f $INSTALLED/sbcl-1.0.30-binary ]
then
    echo -e "\033[32m Downloading sbcl-1.0.30 binary to $DOWNLOAD \033[0m"
    test -f $DOWNLOAD/sbcl-1.0.30-x86-darwin-binary.tar.bz2 || curl -C - -o $DOWNLOAD/sbcl-1.0.30-x86-darwin-binary.tar.bz2 -L http://prdownloads.sourceforge.net
2
    test -d $DOWNLOAD/sbcl-1.0.30-x86-darwin-binary.tar.bz2 || tar -xvf $DOWNLOAD/sbcl-1.0.30-x86-darwin-binary.tar.bz2 -C $DOWNLOAD

    echo -e "\033[36m Installing sbcl-1.0.30 binary in $USR \033[0m"
    cd $DOWNLOAD/sbcl-1.0.30-x86-darwin
    INSTALL_ROOT=$USR/sbcl-1.0.30-binary sh install.sh
    echo -e "\033[36m Installed sbcl-1.0.30 binary sucessfully \033[0m"
    echo `date` >> $INSTALLED/sbcl-1.0.30-binary
    cd -
else
    echo -e "\033[1m sbcl-1.0.30 binary is already installed - skipping \033[0m"
fi

source $HOME/.bash_profile

if [ ! -f $INSTALLED/sbcl-1.0.30 ]
then
    export SBCL_HOME=$USR/sbcl-1.0.30-binary/lib/sbcl

    echo -e "\033[32m Downloading sbcl-1.0.30 from source to $DOWNLOAD \033[0m"
    test -f $DOWNLOAD/sbcl-1.0.30-source.tar.bz2 || curl -C - -o $DOWNLOAD/sbcl-1.0.30-source.tar.bz2 -L http://prdownloads.sourceforge.net/sbcl/sbcl-1.0.30-source
    test -d $DOWNLOAD/sbcl-1.0.30-source.tar.bz2 || tar -xvf $DOWNLOAD/sbcl-1.0.30-source.tar.bz2 -C $DOWNLOAD

    echo -e "\033[36m Installing sbcl-1.0.30 from source in $USR \033[0m"
    cd $DOWNLOAD/sbcl-1.0.30

    cat << EOF > customize-target-features.lisp
(lambda (features)
  (flet ((enable (x)
          (pushnew x features))
        (disable (x)
          (setf features (remove x features)))))
    ;; Threading support, available only on x86/x86-64 Linux, x86 Solaris
    ;; and x86 Mac OS X (experimental).
    (enable :sb-thread)))
EOF

```

My own bash package manager

Yes, the script looks
nasty, but not hard to
maintain (for myself). I
have a record of what all
has been done

I stopped using my own
shell scripts after I found
homebrew



Simple and Clean

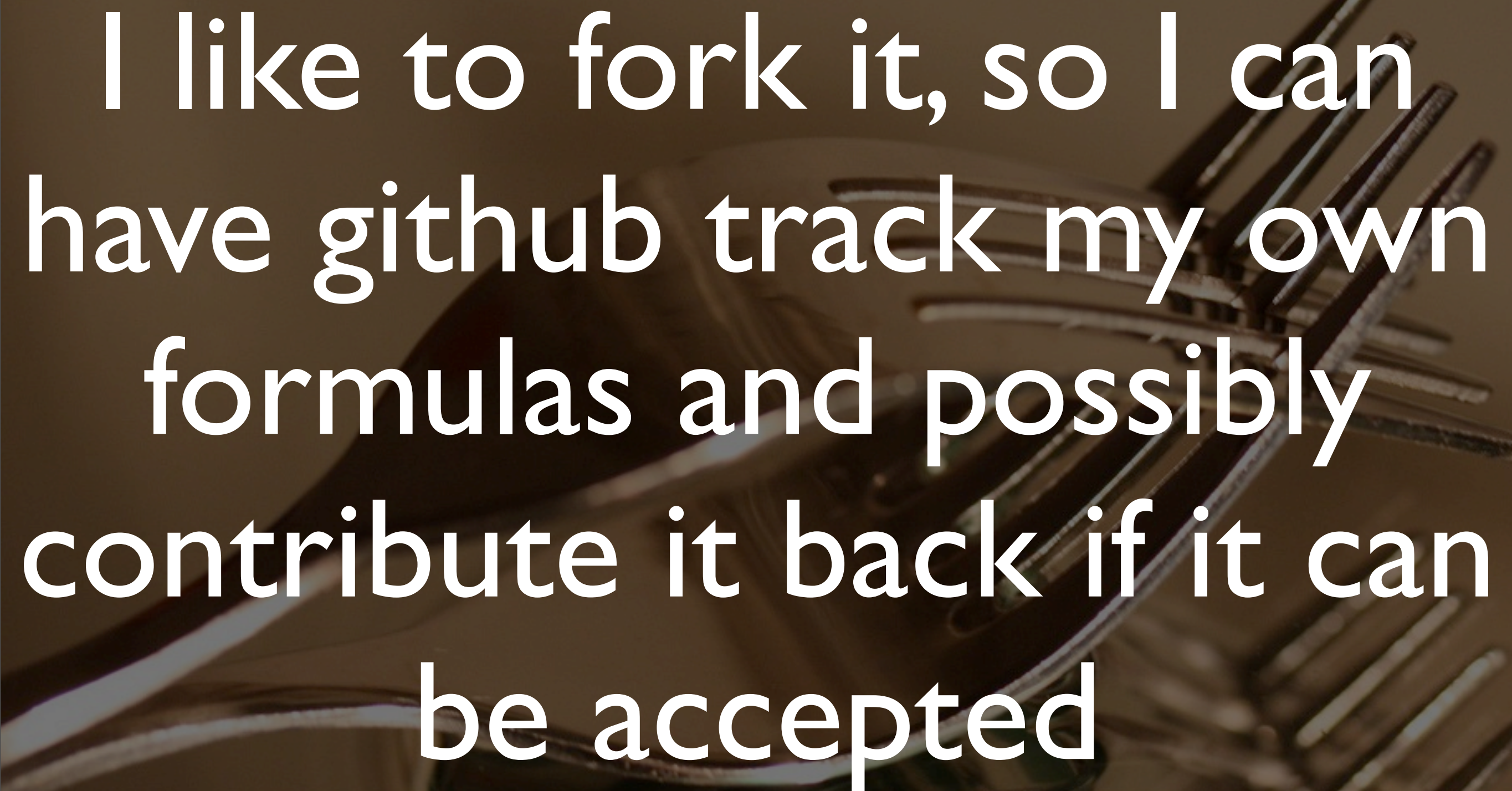
```
brew install git  
brew remove git  
brew list  
brew create http://foo/x.tar.gz
```

Ways to install Homebrew

- Installation Ruby script: <http://gist.github.com/323731>
- `git clone http://github.com/mxcl/homebrew.git`
- `tarball`

The installation script will
install homebrew in /usr/
local, but you can install it
*anywhere!

* Don't forget to add homebrew to PATH if you
installed somewhere else



I like to fork it, so I can
have github track my own
formulas and possibly
contribute it back if it can
be accepted

Requirements

- Working Ruby installation
- Intel CPU
- Snow Leopard
- XCode with XLI*

Homebrew can play with
other package managers
(Macports, Fink), but **be
aware of the risks**

Talk is cheap, let's try to
install something with
homebrew.... How about
MongoDB? :)

```
homebrew/  
|-- Cellar  
|   |-- mongodb  
|   |   |-- 1.4.2-x86_64  
|   |   |   |-- bin  
|   |   |   |-- include  
|   |   |   |   |-- mongo  
|   |   |   |   |   |-- client  
|   |   |   |   |   |-- db  
|   |   |   |   |-- util  
|   |   |-- lib  
|-- Library  
|   |-- Contributions  
|   |-- Formula  
|   |-- Homebrew  
|   |   |-- extend  
|   |   |-- test  
|   |   |   |-- fixtures  
|   |   |   |-- patches  
|-- bin  
|-- include  
|   |-- mongo -> ../Cellar/mongodb/1.4.2-x86_64/include/mongo  
|-- lib  
|-- var  
|   |-- log  
|   |   |-- mongodb  
|   |-- mongodb
```

Don't forget to add bin to your PATH



What if you want an application, but it isn't in homebrew? Create it!

Don't be afraid -
Homebrew is written in
Ruby and so are the
packages (Formulas)

Homebrew Terminology

- Formula - a file that describes the packaged application
- Keg - Installation prefix of a formula
- Cellar - storage location for kegs

How to create a formula?

- Ruby class that represents the formula
- Inherit from Formula
- Specify homepage, url, and md5 within the class
- Implement install method

```

require 'formula'

# iri support requires libidn, which pulls in gettext, so we
# disable this by default.

class Wget <Formula
  homepage 'http://www.gnu.org/software/wget/'
  url 'http://ftp.gnu.org/gnu/wget/wget-1.12.tar.bz2'
  md5 '308a5476fc096a8a525d07279a6f6aa3'

  depends_on "libidn" if ARGV.include? "--enable-iri"

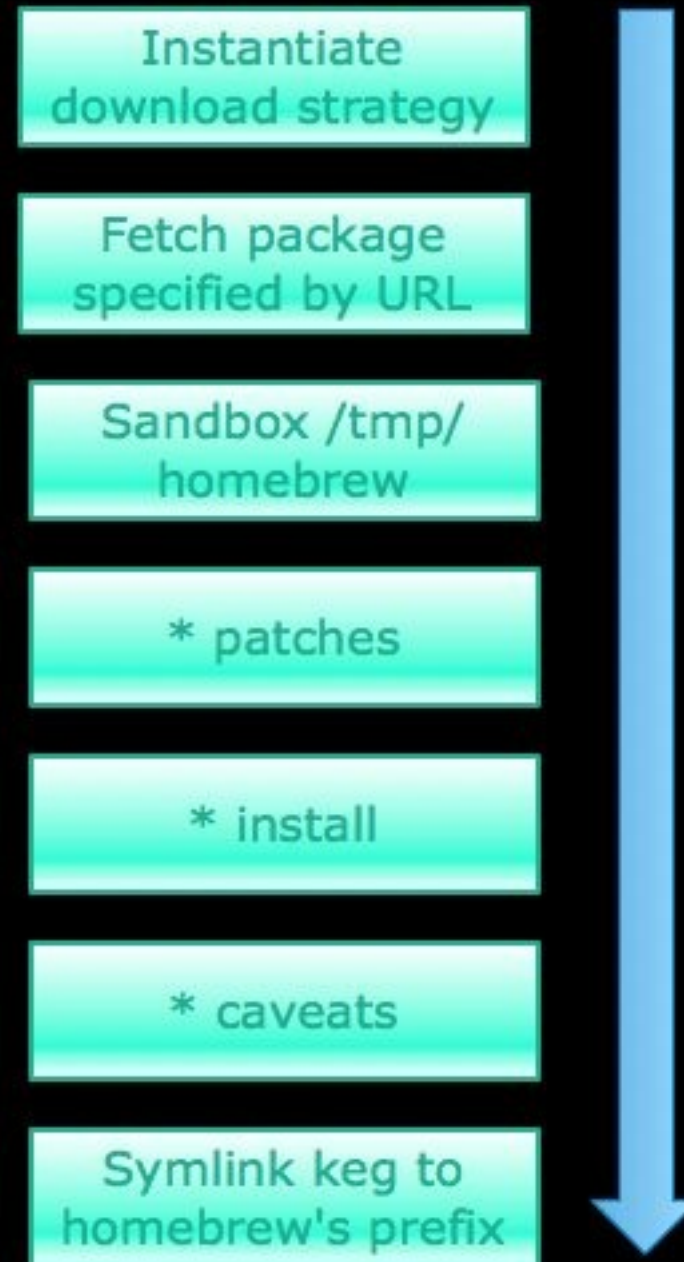
  def install
    args = ["--disable-debug", "--prefix=#{prefix}"]
    args << "--disable-iri" unless ARGV.include? "--enable-iri"

    system "./configure", *args
    system "make install"
  end
end

~
~
~
~
~
~

```

Life cycle



Download Strategy

- curl (rar, jar, zip, gzip, bz2)
- svn, git, cvs, hg, baazar

Homebrew formula types

- ScriptFileFormula
- GithubGistFormula

brew install -vd foo

Formula class contains
same information as my
hand rolled script, but a
lot less duplication

Contribute the formula
back by fork, push, and
create a new issue on the
wiki

What about fink and macports?

```
phiafeng-mb:~ afeng$ port install grep
MacPorts running without privileges. You may be unable to complete certain actions
---> Computing dependencies for grep
MacPorts running without privileges. You may be unable to complete certain actions
MacPorts running without privileges. You may be unable to complete certain actions
---> Fetching expat
---> Attempting to fetch expat-2.0.1.tar.gz from http://voxel.dl.sourceforge.net
MacPorts running without privileges. You may be unable to complete certain actions
---> Verifying checksum(s) for expat
MacPorts running without privileges. You may be unable to complete certain actions
---> Extracting expat
MacPorts running without privileges. You may be unable to complete certain actions
MacPorts running without privileges. You may be unable to complete certain actions
---> Configuring expat
MacPorts running without privileges. You may be unable to complete certain actions
---> Building expat
MacPorts running without privileges. You may be unable to complete certain actions
---> Staging expat into destroot
MacPorts running without privileges. You may be unable to complete certain actions
---> Installing expat @2.0.1_0
Error: Target org.macports.install returned: MacPorts requires root privileges for
Error: The following dependencies failed to build: gettext expat libiconv gperf readline
e bzip2 readline zlib
Error: Status 1 encountered during processing.
Before reporting a bug, first run the command again with the -d flag to get complete
phiafeng-mb:~ afeng$
phiafeng-mb:~ afeng$
```

```
phiafeng-mb:homebrew afeng$ ./bin/brew install grep
Warning: It appears you have Macports or Fink installed
Although, unlikely, this can break builds or cause obscure runtime issues.
If you experience problems try uninstalling these tools.
==> Downloading http://ftp.gnu.org/pub/gnu/gettext/gettext-0.17.tar.gz

==> Downloading http://mirror.its.uidaho.edu/pub/gnu/grep/grep-2.5.4.tar.bz2
##### 100.0%
==> ./configure --disable-debug --disable-dependency-tracking --disable-nls --pre
ew/Cellar/grep/2.5.4 --infodir=/Users/afeng/tmp/homebrew/Cellar/grep/2.5.4/share/
==> make
==> make install
Warning: /Users/afeng/tmp/homebrew/bin is not in your PATH
You can amend this by altering your ~/.bashrc file
==> Summary
/Users/afeng/tmp/homebrew/Cellar/grep/2.5.4: 10 files, 548K, built in 13 seconds
```

5/1/10	Homebrew	Macports	Fink
MongoDB	1.4.2	1.4.1	???
Erlang	R13B04	R13B04	12B-0-1
Cassandra	0.6.1	0.6.0	???
Bash	4.2.1	4.1.5	4.0-33.1
MySQL	5.1.46	5.1.46	5.0.45-2
Ruby	1.9.1-p378	1.9.1-p378	1.8.7-p174

homebrew is still young

5/1/10

Homebrew: 744 formulas

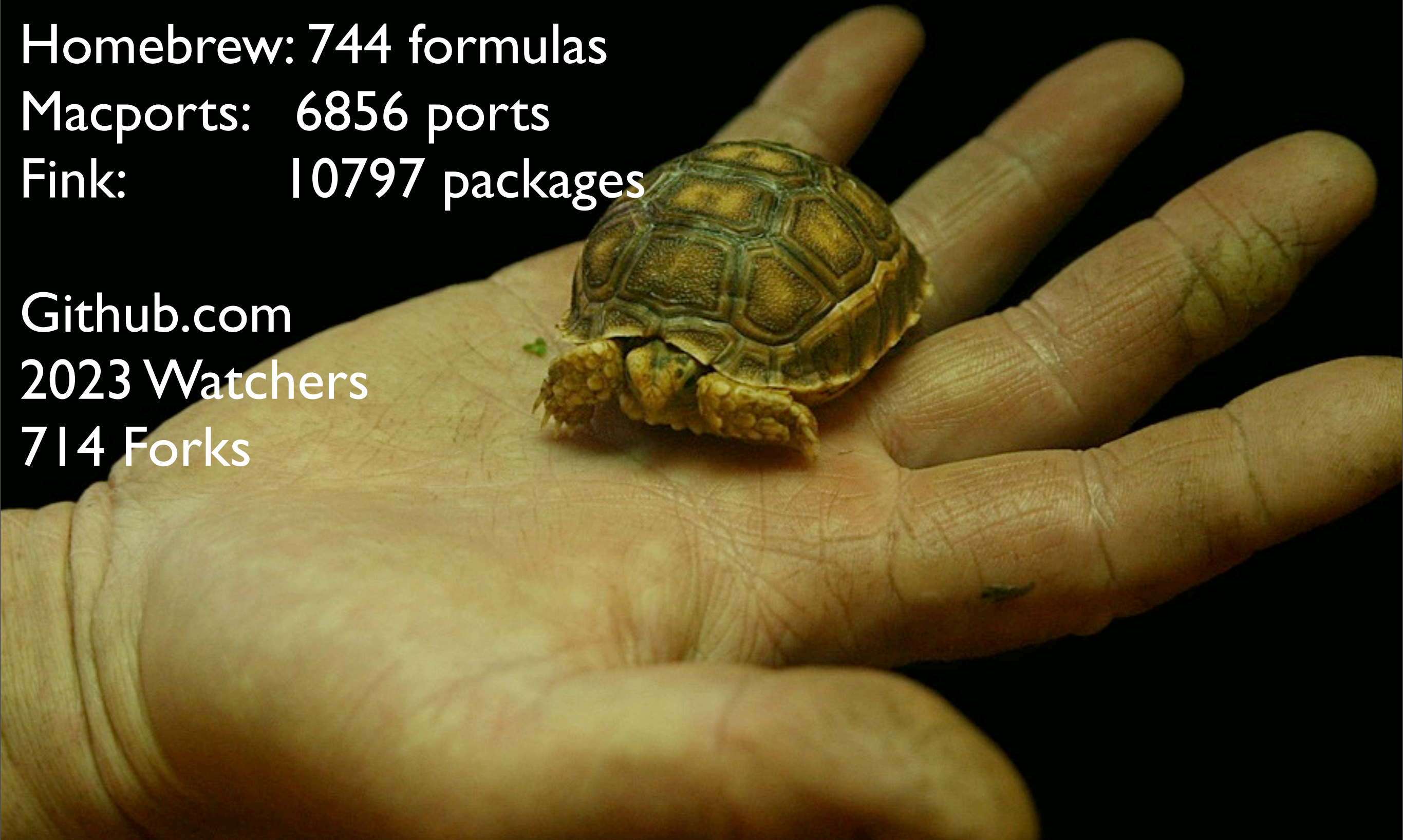
Macports: 6856 ports

Fink: 10797 packages

Github.com

2023 Watchers

714 Forks



Give Homebrew a try. If
you don't like it, just `rm`
`homebrew directory`, and
you are DONE!

Useful homebrew commands

- `brew doctor`
- `brew update`
- `brew home`
- `brew install -i`

Questions?

email:

aaron.feng@gmail.com

twitter:

@aaronfeng

Resources

- Homebrew: <http://mxcl.github.com/homebrew/>
- IRC - Freenode
#machomebrew

Credits

- <http://www.flickr.com/photos/aldoaldoz/2328737233/>
- <http://www.flickr.com/photos/jimbl/2920959955/>
- <http://www.flickr.com/photos/hyakuhei/2234810476/>
- <http://www.flickr.com/photos/michaelwm25/2523941399/>
- <http://www.flickr.com/photos/albertopveiga/3846929292/>
- <http://www.flickr.com/photos/rutty/354366312/>
- <http://www.flickr.com/photos/29487767@N02/2866453076/>