

## CodeSnippets

9881 users tagging and storing useful source code snippets

### Never been to CodeSnippets before?

**Snippets is a public source code repository.** Easily build up your personal collection of code snippets, categorize them with tags / keywords, and share them with the world (or not, you can keep them private!)

## Compile & install cdecl on Mac OS X ([See related posts](#))

```
# 1. readline

# first compile and install the latest version of readline (into /usr/local/lib)
open http://tiswww.case.edu/php/chet/readline/rltop.html

# let's make sure we use standard system libs & files
sudo mv -i /opt/local /opt/local-off
sudo mv -i /usr/local /usr/local-off

#sudo mv -i /opt/local-off /opt/local
#sudo mv -i /usr/local-off /usr/local

cd ~/Desktop

fwftpd # open ipfw firewall for ftp (cf. http://codesnippets.joyent.com/posts/show/1284)

curl -L -O ftp://ftp.cwru.edu/pub/bash/readline-6.1.tar.gz

fwdef # restore default ipfw rules

tar -xzf readline-6.1.tar.gz

cd readline-6.1

./configure

make

sudo mv -i /usr/local-off /usr/local

sudo make install

sudo mv -i /opt/local-off /opt/local

find ~/Desktop/readline-6.1 -name "*dylib"
ls -l /usr/local/lib/*readline*

[[ -d /usr/local ]] && sudo mkdir -p /usr/local/src || echo 'mkdir failed!'

sudo cp ~/Desktop/readline-6.1.tar.gz /usr/local/src

tar -C /usr/local/src -xzf ~/Desktop/readline-6.1.tar.gz

find /usr/local/src/readline-6.1 -name "*compat.c"

otool -L /usr/local/lib/*readline*6*

# 2. cdecl
```

```

# compile & install cdecl (into /usr/local/bin; with readline support)

open http://www.cdecl.org

cd ~/Desktop

curl -L -O http://cdecl.org/files/cdecl-blocks-2.5.tar.gz

tar -xzf cdecl-blocks-2.5.tar.gz

cd cdecl-blocks-2.5

# edit Makefile

sed -i "" -E \
-e 's/^( *CFLAGS *=.+)/\1 -Wall -Ddebug -Ddoyydebug -DUSE_READLINE/' \
-e 's|^ *LIBS *=.+|LIBS= -L/usr/local/lib -lreadline -L/usr/lib -lncurses|' \
-e 's|^ *BINDIR *=.+|BINDIR= /usr/local/bin|' \
-e 's|^ *MANDIR *=.+|MANDIR= /usr/local/share/man/man1|' \
-e 's|^ *CATDIR *=.+|CATDIR = /usr/local/share/man/cat1|' \
Makefile

# edit cdgram.y

cat <<-'EOF' | sed -e 's/^ */' -e 's/ */' | /bin/ed -s cdgram.y
H
li

%{
int yylex(void);
%}

.
wq
EOF

# edit cdecl.c

# replace "Function" by "rl_compentry_func_t"
# while (...) -> while ((...))
# add missing parentheses: if ... { ... }

sed -i "" -E \
-e 's/^( *rl_completion_entry_function *= *)( *Function *)(* *)keyword_completion;)/\1 rl_compentry_func_t \3/' \
-e 's/while +(command += +commands\[index\]) +{/while ((command = commands\[index\])) {/' \
-e 's/while +(option += +options\[index\]) +{/while ((option = options\[index\])) {/' \
-e 's/} +else +while +(keyword += +keywords\[index\]) +{/} else while ((keyword = keywords\[index\])) {/' \
-e 's/^(\[:space:\])+(if \&\& \&(strcmp(type, "void") == 0)))/\1 {/' \
-e 's/^(\[:space:\])+(if \&(strcmp(type, "void") == 0)))/\1 {/' \
-e 's/^(\[:space:\])+"pointer to void";)/\1 }/' \
cdecl.c

# add some include statements to cdecl.c

cat <<-'EOF' | sed -e 's/^ */' -e 's/ */' | /bin/ed -s cdecl.c
H
/^(\[:space:\])\{0,\}#(\[:space:\])\{0,\}include(\[:space:\])\{0,\}<readline/readline.h>/a

#include </usr/local/src/readline-6.1/compat.c>
#include </usr/local/include/readline/readline.h>
#include </usr/local/include/readline/history.h>
#include <unistd.h>
// #include </usr/local/include/readline/rltypedefs.h>
int dotmpfile_from_string(char *s);

.
/^(\[:space:\])\{0,\}#(\[:space:\])\{0,\}include(\[:space:\])\{0,\}<readline/readline.h>/d
wq
EOF

# get rid of warning: missing braces around initializer (in cdecl.c)

cat <<-'EOF' | sed -e 's/^ */' -e 's/ */' | /bin/ed -s cdecl.c

```

```

H
/^[[:space:]]\{0,\}char crosscheck\[9\]\[9\] = {/i

char crossXYZcheck[9][9] = {
/*      L, I, S, C, V, U, S, F, D, */
/* long */      {_, _, _, _, _, _, _, _, _},
/* int */      {_, _, _, _, _, _, _, _, _},
/* short */     {X, _, _, _, _, _, _, _, _},
/* char */     {X, X, X, _, _, _, _, _, _},
/* void */     {X, X, X, X, _, _, _, _, _},
/* unsigned */ {R, _, R, R, X, _, _, _, _},
/* signed */   {P, P, P, P, X, X, _, _, _},
/* float */    {A, X, X, X, X, X, X, _, _},
/* double */   {P, X, X, X, X, X, X, X, _}
};

.
/^[[:space:]]\{0,\}char crosscheck\[9\]\[9\] = {/,+12d
g/crossXYZcheck/s//crosscheck/
wq
EOF

```

```
# open -e cdecl.c Makefile
```

```
make
```

```
make test
```

```
echo $?
```

```
sudo make install
```

```
ls -l "$(type -P cdecl)"
otool -L "$(type -P cdecl)"
type -a c++decl cdecl
ls -l $(type -P c++decl cdecl)
```

```
man cdecl
cdecl --help
```

```

cdecl
explain int ((*foo)(void ))[3]
declare bar as volatile pointer to array 64 of const int
cast foo into block(int, long long) returning double
explain char ** const * const x
declare x as const pointer to const pointer to pointer to char
[ctrl-c]

```

to [shell osx mac bash unix linux c readline cdecl learn](#) by [jvs](#) on Mon Feb 08 19:12:19 +0000 2010

You need to [create an account](#) or log in to post comments to this site.

## Related Posts

- » [Find largest files on Mac OS X](#) in [find awk sort regex shell osx mac bash unix linux c filesystem file ls array mount sed size directory mdfind scan device gnu volume largest printf stat xdev bsd partition bigfiles localvols](#)
- » [Counting files](#) in [find shell osx mac bash unix linux c count file recursive api directory scan manager print0 fts countfiles carbon](#)
- » [Using this function you will...](#) in [apache php mysql find ruby rails time date perl regex shell http osx mac css python google html bash error windows sql activerecord javascript java rubyonrails unix linux database c text csharp XSLT image string xml file array web url math mongrel\\_cluster hash convert dscl C++ sips series60 rexml REBOL jonas jsfromhell raoni sinatra rscript](#)
- » [Statement Emulation](#) in [apache php mysql find ruby rails time date perl regex shell http osx mac css python google html bash error windows sql activerecord javascript java rubyonrails unix linux database c text csharp XSLT image string xml file array web url math mongrel\\_cluster hash convert dscl C++ sips series60 rexml REBOL jonas jsfromhell raoni sinatra rscript](#)
- » [Fill data into an XML templa...](#) in [apache php mysql find ruby rails time date perl regex shell http osx mac css python](#)

[google](#) [html](#) [bash](#) [error](#) [windows](#) [sql](#) [activerecord](#) [javascript](#) [java](#) [rubyonrails](#) [unix](#) [linux](#) [database](#) [c](#) [text](#) [csharp](#) [XSLT](#)  
[image](#) [string](#) [xml](#) [file](#) [array](#) [web](#) [url](#) [math](#) [mongrel](#) [cluster](#) [hash](#) [convert](#) [dsc](#) [C++](#) [sips](#) [series60](#) [rexml](#) [REBOL](#) [jonas](#)  
[jsfromhell](#) [raoni](#) [sinatra](#) [rscript](#)

» [Wrap all methods \(functions\)...](#) in [apache](#) [php](#) [mysql](#) [find](#) [ruby](#) [rails](#) [time](#) [date](#) [perl](#) [regex](#) [shell](#) [http](#) [osx](#) [mac](#) [css](#) [python](#)  
[google](#) [html](#) [bash](#) [error](#) [windows](#) [sql](#) [activerecord](#) [javascript](#) [java](#) [rubyonrails](#) [unix](#) [linux](#) [database](#) [c](#) [text](#) [csharp](#) [XSLT](#)  
[image](#) [string](#) [xml](#) [file](#) [array](#) [web](#) [url](#) [math](#) [mongrel](#) [cluster](#) [hash](#) [convert](#) [dsc](#) [C++](#) [sips](#) [series60](#) [rexml](#) [REBOL](#) [jonas](#)  
[jsfromhell](#) [raoni](#) [sinatra](#) [rscript](#)



Snippets ([source code soon to be available](#)) developed by [Peter Cooper](#) and powered by [Ruby On Rails](#)